



# Developing Network Visualization Tools Using WPF

**Felipe Kober**  
*PowerSysLab - Brazil*

1. What is WPF?
2. Why WPF?
3. Examples
4. Conclusions

## 1. What is WPF?

## 2. Why WPF?

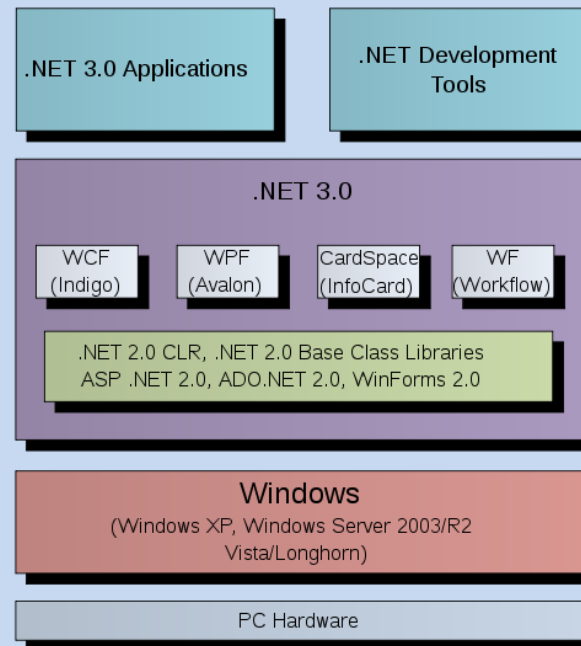
## 3. Examples

## 4. Conclusions

### Windows Presentation Foundation

- Part of .NET Framework
- Graphical subsystem for 2D and 3D Windows based user interfaces
- Alternative for traditional technologies (GDI, Open GL...)

### .NET 3.0 Stack



1. What is WPF?
2. Why WPF?
3. Examples
4. Conclusions

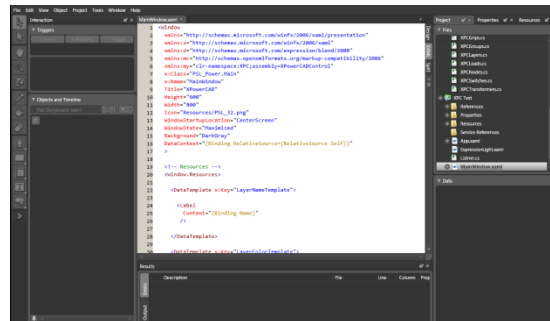
## DESIGN



### XAML

(eXtensible Application Markup Language)

- ✓ XML-based language
- ✓ Defines UI elements, data binding and events
- ✓ Dedicated design tools



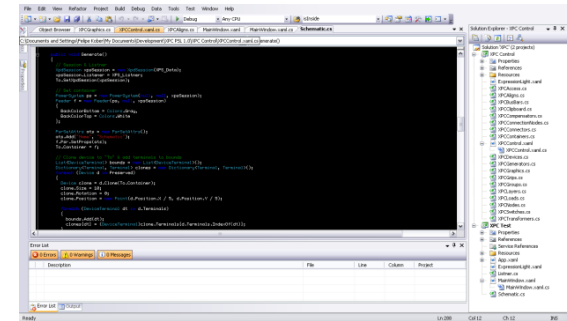
Microsoft Expression Blend

## IMPLEMENTATION



### C#, VB

- ✓ Defines the business logic of the application
- ✓ Dedicated development tools



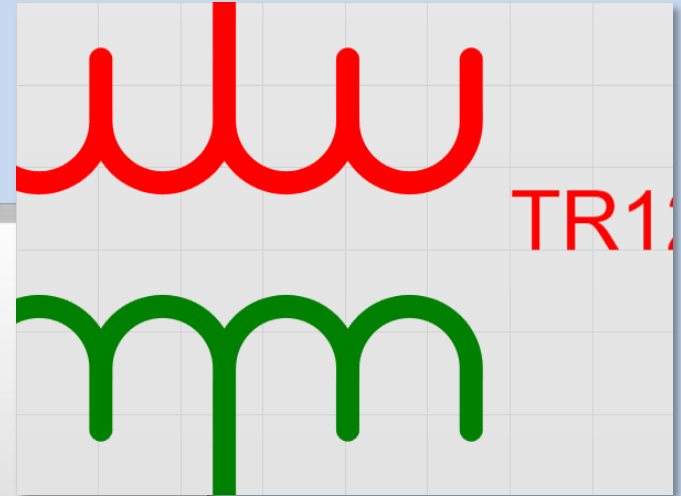
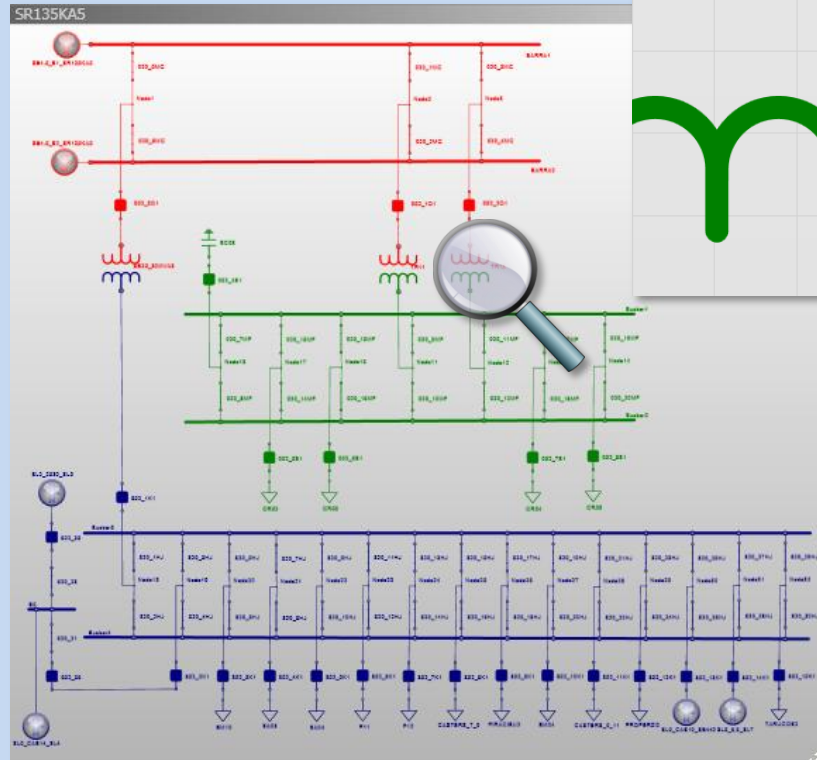
Microsoft Visual Studio



1. What is WPF?
2. Why WPF?
3. Examples
4. Conclusions

## Rendering

- ✓ Vector-based
- ✓ Device Independent Unit



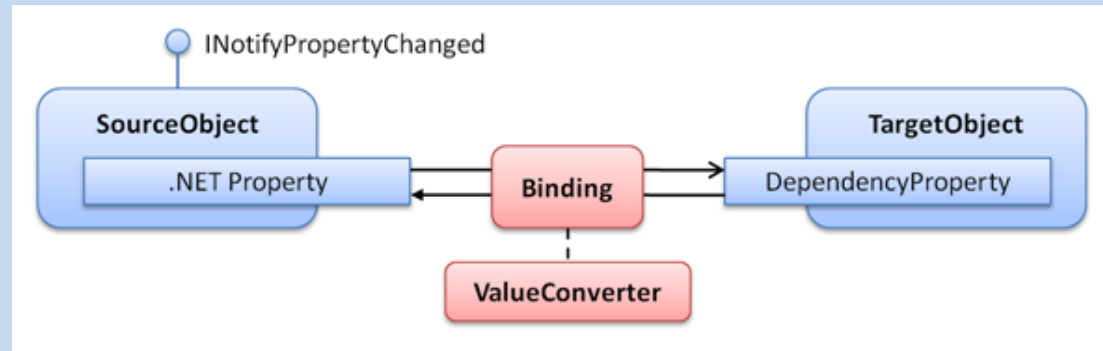
## 1. What is WPF?

## 2. Why WPF?

## 3. Examples

## 4. Conclusions

### Data Binding

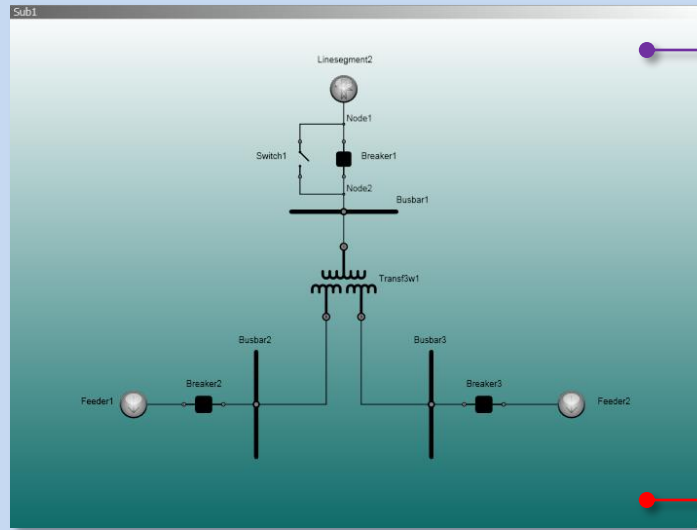


- ✓ Establishes a connection between the application UI and business logic.
- ✓ Target elements automatically reflect changes on source elements data and vice-versa.
- ✓ Less code needed.

1. What is WPF?
2. Why WPF?
3. Examples
4. Conclusions

## Data Binding Background Colors

WPF UI element (Canvas)



C# class

```

XPowerCADCtrl
...
+ BackColorTop : Color
+ BackColorBottom : Color
...
    
```

```

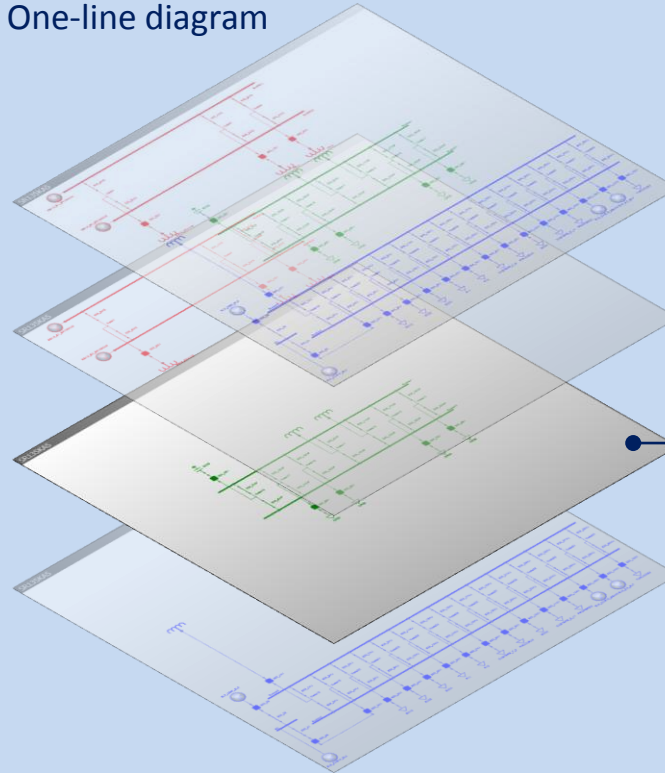
<Canvas.Background>
  <LinearGradientBrush StartPoint="0.5,0" EndPoint="0.5,1">
    <GradientStop Color="{Binding ElementName=XPowerCADCtrl, Path=BackColorTop}" Offset="0.0"/>
    <GradientStop Color="{Binding ElementName=XPowerCADCtrl, Path=BackColorBottom}" Offset="1.0"/>
  </LinearGradientBrush>
</Canvas.Background>
    
```

XAML code

- 1. What is WPF?
- 2. Why WPF?
- 3. Examples
- 4. Conclusions

## Data Binding Voltage Layers

One-line diagram



List View

Layer	Color	Visible
88 kV	<span style="color: blue;">■</span>	<input checked="" type="checkbox"/>
32 kV	<span style="color: green;">■</span>	<input checked="" type="checkbox"/>
230 kV	<span style="color: red;">■</span>	<input checked="" type="checkbox"/>
6,6 kV	<span style="color: blue;">■</span>	<input checked="" type="checkbox"/>
13,8 kV	<span style="color: magenta;">■</span>	<input checked="" type="checkbox"/>
104 kV	<span style="color: cyan;">■</span>	<input checked="" type="checkbox"/>
115 kV	<span style="color: green;">■</span>	<input checked="" type="checkbox"/>
120,5 kV	<span style="color: lightblue;">■</span>	<input checked="" type="checkbox"/>
440 kV	<span style="color: darkred;">■</span>	<input checked="" type="checkbox"/>

Template

115 kV ■

```

VoltageLayer
...
+ Voltage: double
+ Color: Color
+ Visible: bool
...
    
```

C# class

## 1. What is WPF?

## 2. Why WPF?

## 3. Examples

## 4. Conclusions

## Data Binding

### *Voltage Layers*

```

<!-- Layers -->
<ListView
  Name="LayersListView"
  SelectionMode="Single"
  BorderThickness="0"
  FontSize="10"
  ItemsSource="{Binding ElementName=XPowerCAD, Path=Layers}"
  Margin="4,0,0,4"
  HorizontalAlignment="Stretch"
>

  <ListView.View>

    <GridView>

      <GridViewColumn
        Width="100"
        Header="Layer"
        CellTemplate="{StaticResource LayerNameTemplate}"
      />

      <GridViewColumn
        Width="50"
        Header="Color"
        CellTemplate="{StaticResource LayerColorTemplate}"
      />

      <GridViewColumn
        Width="50"
        Header="Visible"
        CellTemplate="{StaticResource LayerVisibleTemplate}"
      />

    </GridView>

  </ListView.View>
</ListView>

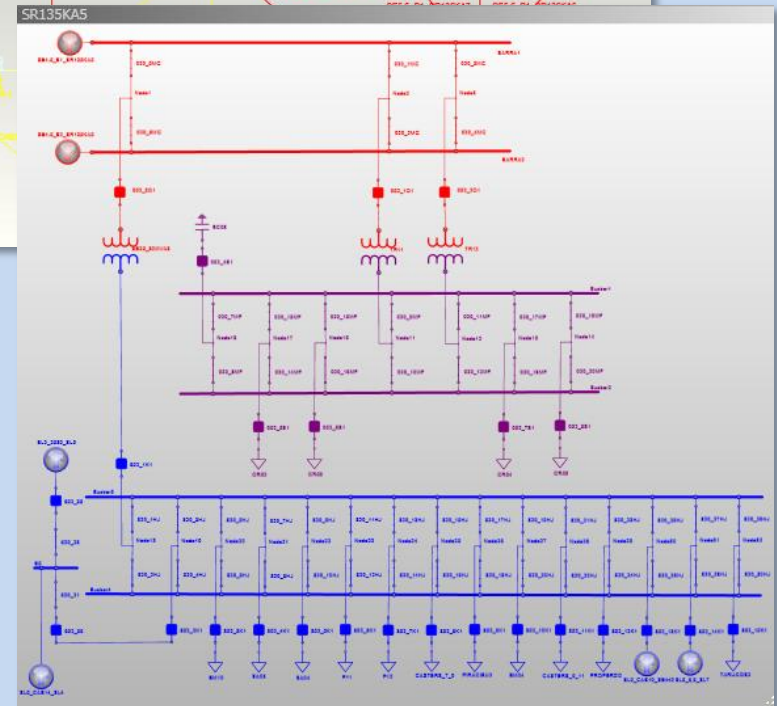
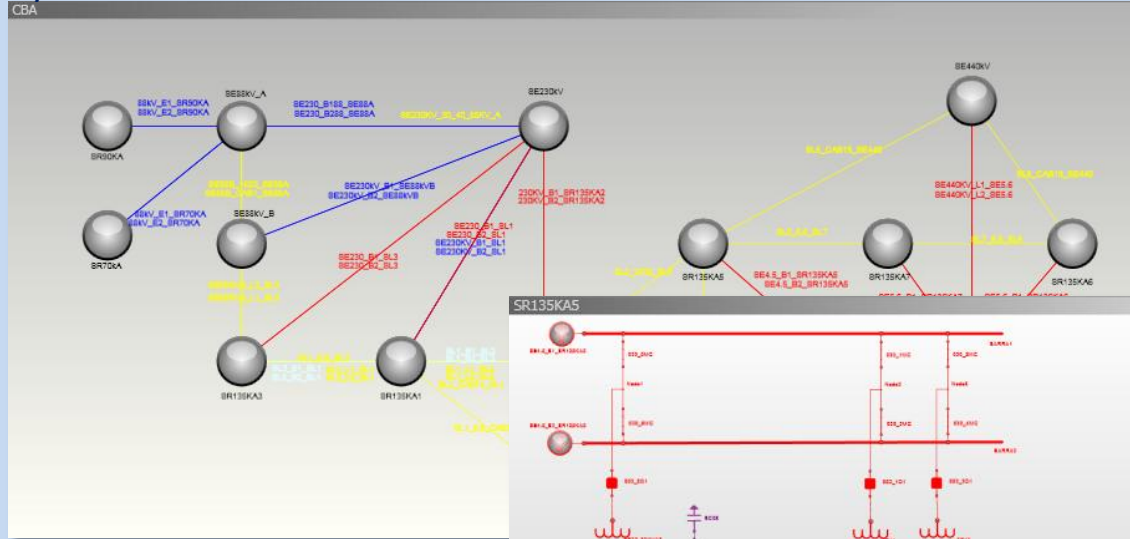
```

XAML code

1. What is WPF?
2. Why WPF?
3. Examples
4. Conclusions

## Industrial Plant

System View

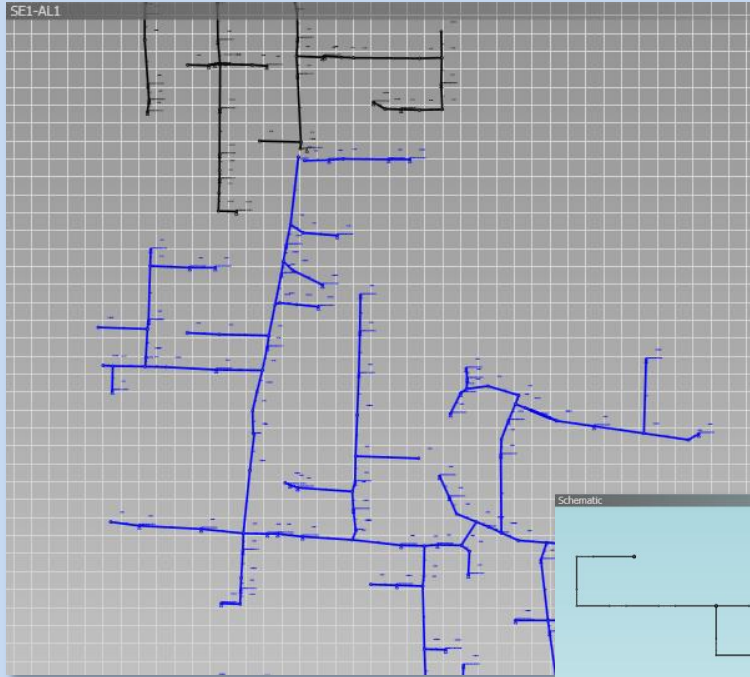


Substation View

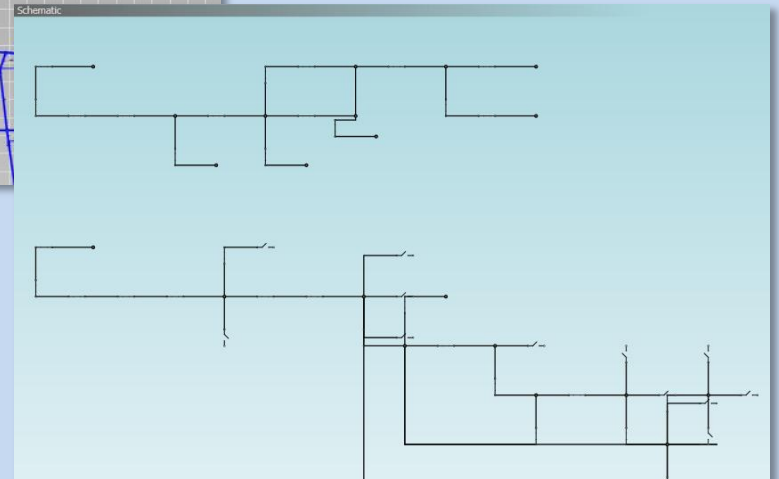
1. What is WPF?
2. Why WPF?
3. Examples
4. Conclusions

## Distribution

Geographic View



Schematic View



## 1. What is WPF?

## 2. Why WPF?

## 3. Examples

## 4. Conclusions

- ✓ Consistent programming model
- ✓ Rendering engine
- ✓ Overall development cost and time reduced
  - Less code
  - Design X Implementation
  - Less maintenance effort



**THANK YOU**

<INFO name="Felipe Kober" email="felipe.kober@powersyslab.com"/>