A Platform for the Development of Mathematical games on Silverlight

mr Davorka Radaković
dr Đorđe Herceg
Department of Mathematics and Informatics
Faculty of Sciences, Novi Sad

CADGME 2012, Novi Sad, Serbia

What is “SLGeometry”

- Developed at the Faculty of Sciences, Novi Sad
- Stands for “Silverlight Little Geometry”
- A platform for developing dynamic geometry applications, interactive math games, demonstrations, teaching materials etc.
- Open-source project, under development, C# project
- Runs as a Silverlight applet or as a desktop application
- Interpreted functional input language
- Visual objects are represented by functions
- Dynamic evaluation of functions causes dynamic screen updates → dynamic geometry!

References

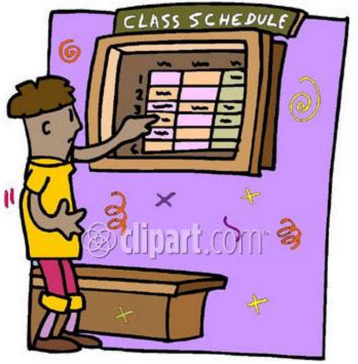
- D. Radaković, Đ. Herceg, **Proširiva modularna platforma za dinamičku geometriju**, 12 Srpski matematički kongres, Zbornik radova – Sekcija VI, Novi Sad (2008), pp. 185-194

- D. Radaković, Đ. Herceg, **The Use of WPF for Development of Interactive Geometry Software**, Acta Univ. M. Belii ser. Mathematics 16 (2010), pp. 65-79

- D. Radaković, Đ. Herceg, M. Löberbauer: **Extensible expression evaluator for the dynamic geometry software Geometrijica**, PRIM 2009, Novi Sad (2010), pp. 95-100

- Đ. Herceg ,D. Radaković: **The Extensibility of an Interpreted Language Using Plugin Libraries**, Numerical Analysis and Applied Mathematics ICNAAM 2011, AIP Conf. Proc. 1389 (2011), pp. 837-840

Summary



- Motivation
- Overview of *SLGeometry*
- Visual controls
- Demo
- Conclusion

Motivation

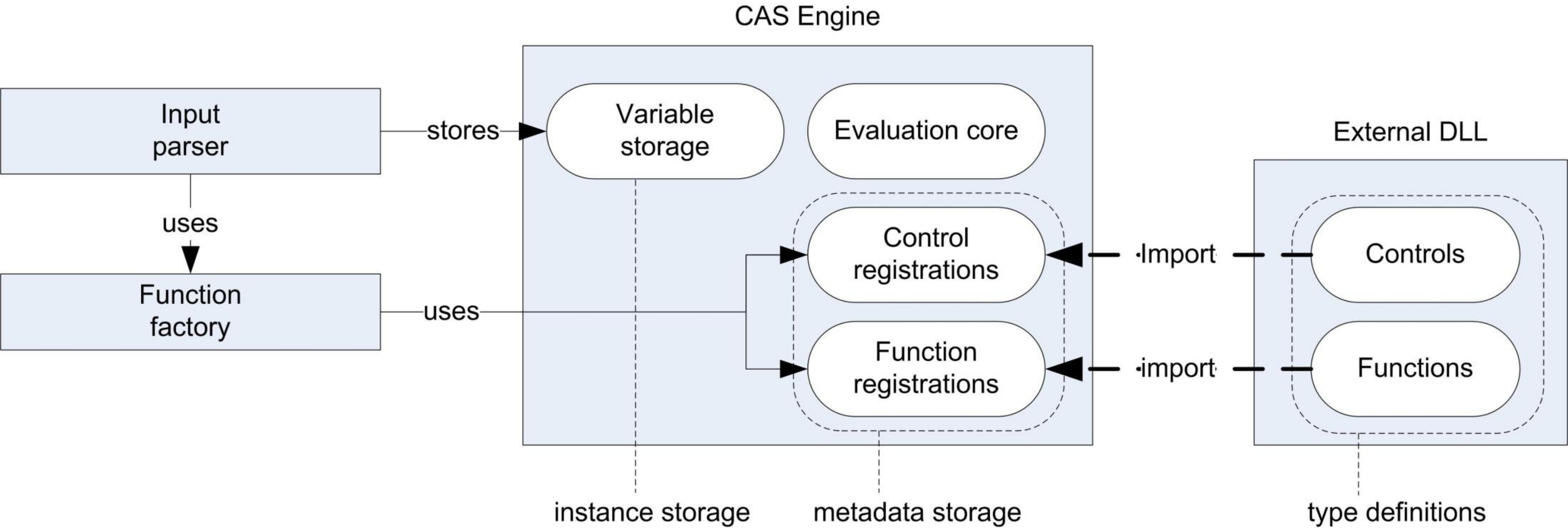
- Develop a component-based dynamic geometry system
- Easy integration into other projects
- Improve some of the best features of other DGSs
- Extend existing expression syntax (particularly from *GeoGebra*) by implementing OO-like syntax
- Simple extensibility model for adding new functions and new visual components
- Import of Silverlight-compatible user controls

Advantages of OO-like syntax

- Improved readability
  - Point.X instead of X(Point)
  - Segment.Midpoint instead of Midpoint(Segment)
- Reduced number of specialized functions
  - Clock.Hour better than Hour(Clock)
- Intuitive – first specify object, then its property
  - Segment.Midpoint → evaluates to a point in plane
  - Segment.Midpoint.X → evaluates to a number
- Analogy to mobile phones:
  - First choose a contact, then phone no., address, etc.

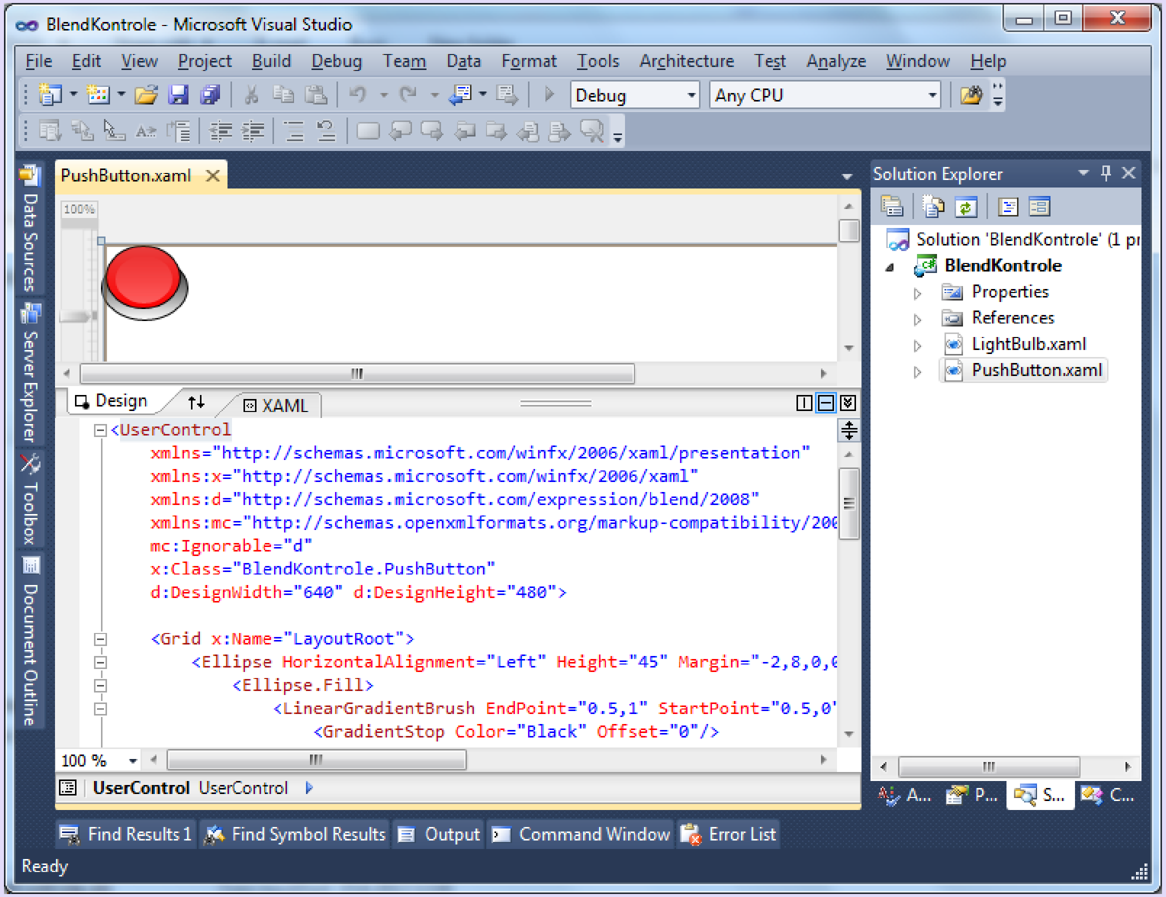
Components of “SLGeometry”

- Parser – Coco/R
- Expression evaluator
- Graphical subsystem (GeoCanvas + visual objects)
- External functions and Controls from DLL files



Authoring a visual control

- Design the new control in Expression Blend or Visual Studio
- Compile the DLL
- Copy the DLL file to the Web Server

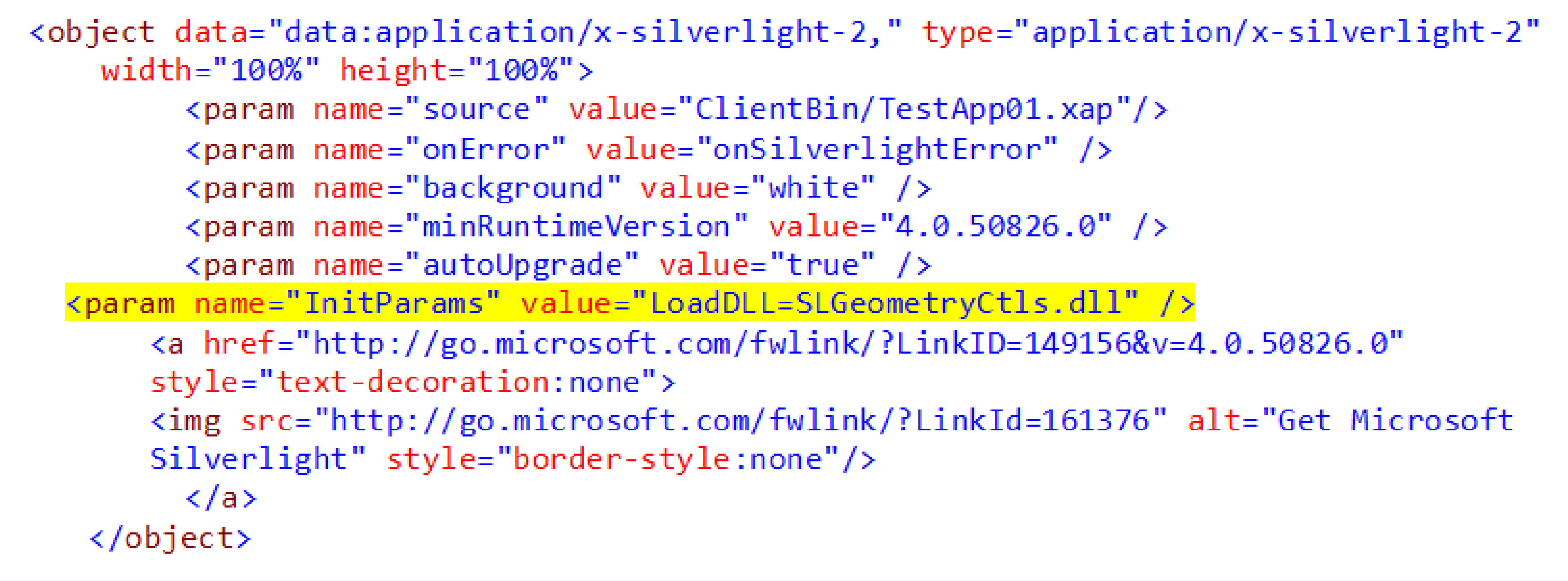


```
<UserControl
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/200
    mc:Ignorable="d"
    x:Class="BlendKontrole.PushButton"
    d:DesignWidth="640" d:DesignHeight="480">

    <Grid x:Name="LayoutRoot">
        <Ellipse HorizontalAlignment="Left" Height="45" Margin="-2,8,0,
            <Ellipse.Fill>
                <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0
                    <GradientStop Color="Black" Offset="0"/>
```

Importing the DLL

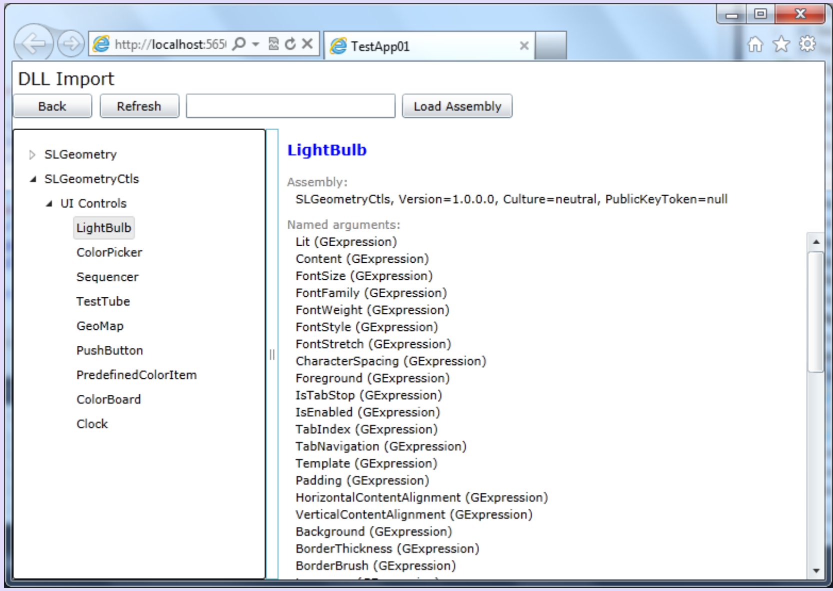
- Specify the DLL(s) to be imported in the Web page that runs SLGeometry applet



```
<object data="data:application/x-silverlight-2," type="application/x-silverlight-2"
    width="100%" height="100%">
         <param name="source" value="ClientBin/TestApp01.xap"/>
         <param name="onError" value="onSilverlightError" />
         <param name="background" value="white" />
         <param name="minRuntimeVersion" value="4.0.50826.0" />
         <param name="autoUpgrade" value="true" />
  <param name="InitParams" value="LoadDLL=SLGeometryCtls.dll" />
       <a href="http://go.microsoft.com/fwlink/?LinkID=149156&v=4.0.50826.0"
       style="text-decoration:none">
       <img src="http://go.microsoft.com/fwlink/?LinkId=161376" alt="Get Microsoft
       Silverlight" style="border-style:none"/>
         </a>
   </object>
```

Using the imported controls

- New functions are defined automatically for all imported controls
- Properties of controls are mapped to the named “properties” of the functions
- Dynamic binding is possible -> animation!



DLL Import

- SLGeometry
- SLGeometryCtls
  - UI Controls
    - LightBulb
    - ColorPicker
    - Sequencer
    - TestTube
    - GeoMap
    - PushButton
    - PredefinedColorItem
    - ColorBoard
    - Clock

LightBulb

Assembly:
SLGeometryCtls, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null

Named arguments:
Lit (GExpression)
Content (GExpression)
FontSize (GExpression)
FontFamily (GExpression)
FontWeight (GExpression)
FontStyle (GExpression)
FontStretch (GExpression)
CharacterSpacing (GExpression)
Foreground (GExpression)
IsTabStop (GExpression)
IsEnabled (GExpression)
TabIndex (GExpression)
TabNavigation (GExpression)
Template (GExpression)
Padding (GExpression)
HorizontalContentAlignment (GExpression)
VerticalContentAlignment (GExpression)
Background (GExpression)
BorderThickness (GExpression)
BorderBrush (GExpression)

Demo

- Clock
- CheckBox
- PushButton
- Sequencer
- LightBulb
- GeoMap

Benefits

- Rich visual experience
- Interactivity
- Reusability
- Portability (just copy the DLL and the applet)
- Integration with the dynamic expressions
- Components with state enable us to develop “sequential” behavior, in contrast with the “straightforward” dynamic behavior
- Inexperienced users can use advanced components in their drawings!

Disadvantages

- Programming skills needed to develop controls
- Programming tools (Visual Studio, Blend) necessary
- DLL files must be hosted on the Web server

Comparison to GeoGebra tools

- GeoGebra Tools are simpler to use, but not as powerful
- GeoGebra Tools are packed inside .ggb files
- SLGeometry components are in separate .dll files
- GeoGebra Tools create many object each time they are used
- Each SLGeometry component is a single object

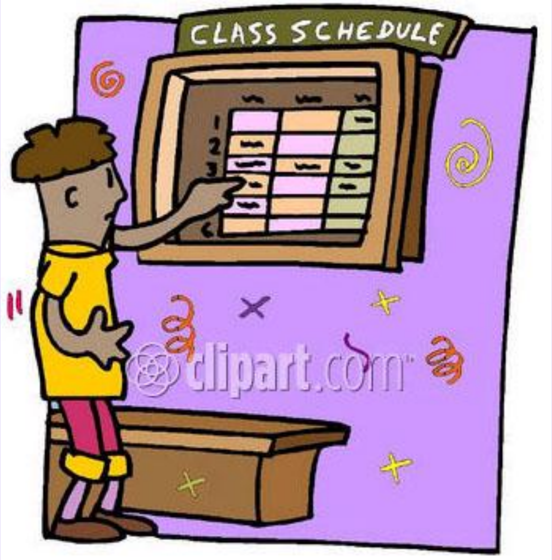
Who will use components?

- Programmers and designers will make them
- Users of SLGeometry will download DLL files and use the components in their drawings
- Project source code (under development)

http://sites.dmi.pmf.uns.ac.rs/personal/hercegdj/geometrijica/

Conclusion

- Our goal was to provide easy extensibility and rich interactive controls
- Functions and visual objects are packed into libraries and registered at runtime
- The result is an modular and easily extensible dynamic geometry software



Thank you!