



STUDENT MONITORING SYSTEM USING WPF AND WEB SERVICES

Dr. Samrat O Khanna

Institute of Sci. & Tech. for Adv Std & Res.

Mijal A Mistry

Institute of Sci. & Tech. for Adv Std & Res.

Abstract: Windows Presentation Foundation makes possible new kinds of user experiences that can't exist in a browser. This paper presents the usefulness of WPF into developing the rich client interface application. It provides us the new way of developing the application. It gives the same look and feel to both designers and developers where as earlier technology doesn't provide such kind of functionality. This paper represents the system of student management using WPF with rich interaction and flexibility. This system also works on intranet so the data which are stored into the system is accessible throughout LAN. This paper also presents the interoperability across network and how it is implemented using WPF. Now a day's people are moving for richer interface and more flexible way for using the system. This goal can be achieved using WPF. Wide varieties of tools are available for designing and developing WPF application which attracts the developer and designer to move on WPF.

Keywords: WPF, Email, Intranet, XAML, CLR, Milcore, GUI

applications are an entirely different way to think about user interface and design. Because of that, WPF is in a position to turn heads and bring people around to realizing how much life is left in the desktop and why the browser is so limiting. [2]

I. INTRODUCTION

The primary WPF programming model is exposed through managed code. Early in the design phase of WPF there were a number of debates about where the line should be drawn between the managed components of the system and the unmanaged ones. The CLR provides a number of features that make development more productive and robust (including memory management, error handling, common type system, etc.) but they come at a cost. [1]

The major components of WPF are illustrated in the figure below. The red sections of the diagram (PresentationFramework, PresentationCore, and milcore) are the major code portions of WPF. Of these, only one is an unmanaged component – milcore. Milcore is written in unmanaged code in order to enable tight integration with DirectX. All display in WPF is done through the DirectX engine, allowing for efficient hardware and software rendering. WPF also required fine control over memory and execution. The composition engine in milcore is extremely performance sensitive, and required giving up many advantages of the CLR to gain performance. [1]

Windows Presentation Foundation really is the next level, and while it's not cross platform, it provides the ability to create experiences that simply cannot exist in the browser. Apollo applications are enhanced browser apps, but WPF

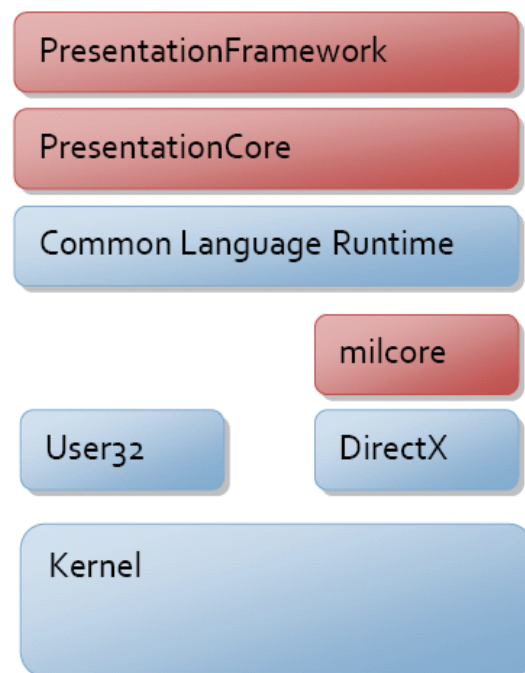


Fig. 1 WPF architecture

WPF is designed to allow you to create dynamic, data driven presentation systems. Every part of the system is designed to create objects through property sets that drive behavior. Data binding is a fundamental part of the system, and is integrated at every layer. [1]

Traditional applications create a display and then bind to some data. In WPF, everything about the control, every aspect of the display, is generated by some type of data binding. The text found inside a button is displayed by creating a composed control inside of the button and binding its display to the button's content property. [1]

When you begin developing WPF based applications, it should feel very familiar. You can set properties, use objects, and data bind in much the same way that you can use Windows Forms or ASP.NET. With a deeper investigation into the architecture of WPF, you'll find that the possibility exists for creating much richer applications that fundamentally treat data as the core driver of the application. [1]

II. FEATURES OF WPF

As several people are saying the why we require to use WPF. Following is the list of features which gives the idea about what can be done using WPF and how it is ahead of other technologies.

New rendering engine

- Retained mode compositing model
- Drawn visuals are cached
- Vector graphics based
- □ Logical pixels
- Transforms, overlays, opacity
- Very different from User32 and GDI32
- DirectX under the covers
- Takes advantage of GPU for graphics processing

For developing the system WPF requires to understand the new concept of XAML. It is a new programming language. XAML stands for eXtensible Application Markup Language. It allows enhanced separation between designer and developer. It is used to define the static structure and configuration of an object Hierarchy say window layout. Here are several other features which is used in XAML. [3]

- Easier to write development tools
- Relatively easy to edit and understand
- Expresses object hierarchies in a more compact form
- Objects must have default constructor
- Requires type conversion

WPF model can be categorized into three different models. They are:

- Code only: WPF objects are all defined as .NET classes. It can instantiate using a programming model similar to Windows Forms. The Design tools will all be based on XAML markup Element hierarchy can be more compactly and cleanly represented in XAML. [3]
- XAML only: It can load raw XAML "pages" into Internet Explorer. It can embed C# or VB code in script blocks within XAML.
- XAML & Code: XAML very expressive for static layout of UI and initial configuration of properties. Programmatic code needed for dynamic behavior of applications. [3]

III. DEVELOPED SYSTEM

Keeping all these things into mind we have developed a browser based system with rich client interface. We have taken the student management and information system for the institute. We made a system which is run in internal network of the institute and allow keeping the database at central location and which also allows getting the updated information whenever it is required. This is important benefit of the system. Sometime it happens that we want to send some information to the students for e.g for sending exam results, some notice for events, some notification, semester break and starting information. To address all these things we have designed several forms within this application. Following code snippet shows the XAML code for registration form. All the controls are placed into the grid as it works as the main container for the different controls into the XAML. In XAML form kind of container is not available so grid, panel act as the container.

For developing WPF application there are several tools available like Expression Blend, Visual Web Developer, Visual Studio etc. Here we have used Visual Studio 2010 for developing the system.

```
<Window x:Class="StudentManagemnet.Window2"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Registration Form" Height="600"
Width="500">
<Grid>
<Label Height="28" Margin="9,11,0,0" Name="label1"
VerticalAlignment="Top" HorizontalAlignment="Left"
Width="71">Name:</Label>
<TextBox Height="23" Margin="101,11,57,0"
Name="textBox1" VerticalAlignment="Top" />
</Grid>
```

Fig.2 shows the registration screen for the student's management information. It accepts the values and stores

it into the database which can be further useful. Fig. 3 shows the email sending screen which is used for sending email for students.



Fig.2 Sample Screen for Student Registration Form

By developing this system we came to know the real power of WPF. Earlier we focused on the traditional asp.net technologies for developing the application. If we want to use the rich look and feel we have to take the help of designer or we have to depend on the third party controls. The use of WPF eliminates this possibilities and WPF itself allow us to developed rich GUI based application and it has inbuilt predefined library for controls so we do not have to depend on other third party controls. Our future work involves providing more secure way for email transmission and also wants to provide the attachments facility with the email, searching email addresses for the students as well as faculty members.

REFERENCES

- [a]<http://msdn.microsoft.com/en-us/library/ms750441.aspx>
- [b] <http://www.zdnet.com/blog/stewart/wp-f-and-proving-the-importance-of-experience/269>
- [c]<http://www.socaldotnetarchitecture.org/Portals/0/Repository/WPFArchitecture.pdf>

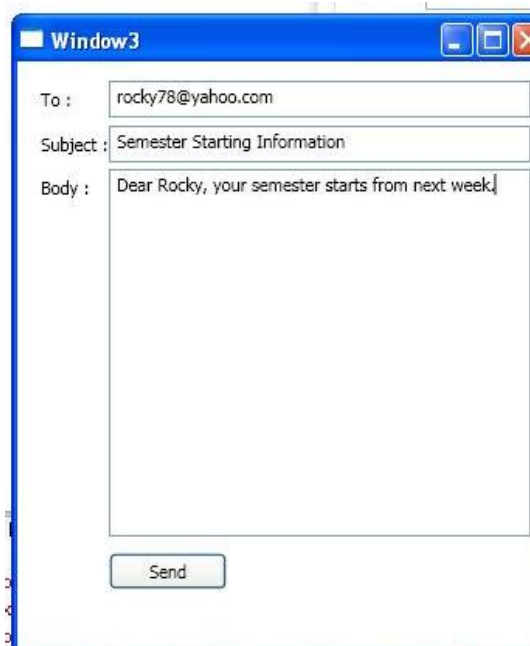


Fig. 3 Email Sending Screen

IV. CONCLUSION