

технологией очень мала, что делает целесообразным построение корпоративной сети информационной системы именно на одной из двух технологий. В процессе выбора в приведенном примере была выбрана технология Wi-Fi ввиду ценового фактора оборудования (даже с учетом разницы в стоимости адаптеров для ПК в пользу Bluetooth) и достаточности допустимого уровня помех. Кроме того, выполнение мультимедийных задач требовало более высокой скорости передачи данных.

Материалы данной работы, изложенные в более детальном и развернутом виде, могут быть приняты

в качестве основы процедуры, позволяющей выполнять выбор вида технологии и комплекса технических средств при проектировании корпоративной сети информационной системы.

Литература

1. <http://www.thg.ru>
2. <http://megalib.com>
3. <http://www.wi-fi.ru>
4. <http://www.bluetooth.ru>

УДК 65.011.56

# ПРИМЕНЕНИЕ ПРОГРАММНЫХ ТЕХНОЛОГИЙ WPF И SILVERLIGHT ПРИ РАЗРАБОТКЕ ИНФОРМАЦИОННЫХ СИСТЕМ

**А.А. Ковальчук**  
Студент\*

Контактный тел.: (057)702-13-37

**Е.П. Павленко**

Кандидат технических наук, доцент

Кафедра информационных управляющих систем\*

Контактный тел.: (057)702-14-51

e-mail: evg-pavl@mail.ru

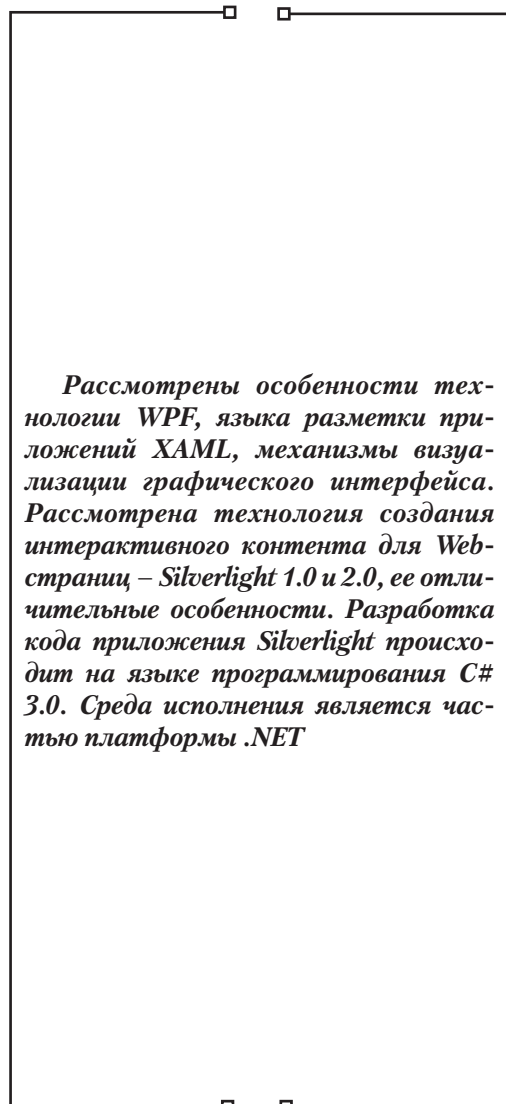
**В.А. Айвазов**

Старший преподаватель

Кафедра охраны труда\*

Контактный тел.: (057)7021-451

\*Харьковский национальный университет радиоэлектроники  
просп. Ленина, 14, г. Харьков



*Рассмотрены особенности технологии WPF, языка разметки приложений XAML, механизмы визуализации графического интерфейса. Рассмотрена технология создания интерактивного контента для Web-страниц – Silverlight 1.0 и 2.0, ее отличительные особенности. Разработка кода приложения Silverlight происходит на языке программирования C# 3.0. Среда исполнения является частью платформы .NET*

## 1. Введение

Во второй половине 2006 г. компания Microsoft выпустила платформу Microsoft .NET Framework 3.0. Впервые были представлены революционные технологии с принципиально новым и качественным подхо-

дом к разработке программного обеспечения, которые предлагается использовать при разработке информационных систем.

Платформа Microsoft .NET Framework 3.0 была представлена в виде дополнений к ядру 2-й версии, и тематически была разделена на три части:

- WCF (Windows Communication Foundation);
- WF (Workflow Foundation);
- WPF (Windows Presentation Foundation).

---

## 2. Постановка задачи

---

Основная задача работы – исследование особенностей использования технологий WPF и Silverlight при разработке программного обеспечения Web-базируемых информационных систем.

Основные требования к программному обеспечению Web-базированной информационной системы:

- обеспечение автоматизированного формирования стандартных, используемых в бизнес-процессе организации документов;
- обработка входной информации;
- быстрота и надежность функционирования информационной системы;
- все запросы на сервер должны выполняться только по необходимости;
- выполнение учетных задач;
- структуризация процесса ведения документооборота организации;
- проект программного обеспечения должен выполняться в рамках заданного финансового бюджета.
- выполнение программного проекта должно осуществляться при использовании наследуемого программного обеспечения или компонентов многократного использования.

---

## 3. Технология WPF

---

Исходя из вышеперечисленных требований, предлагается при разработке программного обеспечения использовать технологию WPF.

Специально для технологии WPF фирмой Microsoft разработан XML-базируемый язык разметки приложений – XAML (Extensible Application Markup Language).

Этот язык позволяет разделить описание приложения, его интерфейсов и ресурсов от их реализаций, тем самым выделяя два этапа разработки – дизайн и реализация, которые могут проводится параллельно. Удобство и простота XAML при разработке интерфейса пользователя Web-базированной информационной системы проста, интуитивно понятна, и очень напоминает написание HTML-страниц, но отличительной чертой XAML является то, что им можно описывать не только интерфейс, но и его поведение, а это в свою очередь, позволяет создавать интерактивные и динамичные приложения.

Принципиально новыми были анимации, стили и эффекты для элементов графического интерфейса, шаблоны, позволяющие полностью менять внешний вид элемента.

Также были добавлены новые концепции построения интерфейсов, такие, как, например, страничная навигация в рамках приложения.

Помимо нового языка разметки, для WPF был также создан новый механизм визуализации интерфейсов пользователя, дающий разработчику невиданную до сих пор гибкость и расширяемость.

Дело в том, что сами элементы интерфейса по сути являются полноценными трёхмерными объектами, рендеринг которых осуществляет непосредственно Microsoft DirectX.

Само же ядро WPF по структуре и принципам работы похоже на «движки» современных трёхмерных игр.

Механизм работы следующий: при отображении WPF-окна создается обычное Win32 окно (неявно вызывается функция CreateWindowExW из user32.dll, а также неявно выполняются все предшествующие вызову этой функции, и все последующие ему действия, необходимые для создания и отображения окна). На этом сходство со всеми технологиями создания GUI для Windows (библиотеки WinAPI, MFC, VCL, CLX, OWL, Windows Forms и многие другие) заканчиваются: ядро DirectX выполняет рендеринг контента окна, выделяя для него отдельный контекст рендеринга. Созданное Win32 окно помещается в вышеупомянутый контекст.

Контент, подобно тому, как это происходит в 3-х мерных компьютерных играх, прорисовывается в Back-буфере кадров графического ускорителя. По окончании рендеринга происходит смена (swap) адресов буферов видеокарты (Back на Front Buffer), после чего, GDI+ отображает визуализированный контент в окно.

Вот пример такого приложения (см. пример и рис.1):

---

## 4. Технология Silverlight

---

Silverlight представляет собой технологию создания интерактивного контента для Web-страниц. В основу данной технологии положены минимальные возможности из WPF. Silverlight 1.0-приложение встраивается в Web-страницу, подобно тому, как это происходит в случае с основными его конкурентами – Macromedia Flash, Adobe Flex и теряющими свою актуальность апплетами Java.

Среди достоинств Silverlight 1.0 можно отметить поддержку 3D графики и потокового видео. Ощутимым недостатком было то, что весь программный код приложения, не относящийся к разметке интерфейса, реализовывался на JavaScript с расширенной объектной моделью.

Такой подход не давал разработчикам требуемой функциональности, и построение сколько-нибудь сложной системы не представлялось возможным. Эта версия была ориентирована скорее на дизайнеров, чем на разработчиков.

Кроме того, в таких приложениях не представлялось возможным использовать стандартные элементы управления, вместо этого набор графических элементов ограничен 3D объектами и геометрическими фигурами.

Новая версия Silverlight 2.0, выпущенная в 2008 г., раскрывает огромные возможности для разработчиков программного обеспечения Web-базируемых информационных систем.

Технология Silverlight 2.0 представляет собой средство создания динамического интерактивного

## App.xaml

```
<Application
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  StartupUri="MainWindow.xaml"/>
```

## MainWindow.xaml

```
<Window x:Class="TestWpf.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  Title="TestWpfWindow" Height="145" Width="300">
  <Canvas>
  <Button Canvas.Left="61" Canvas.Top="32"
    Height="48" Width="157"
    Click="button_Click">
    Simple WPF Button
  </Button>
  </Canvas>
</Window>
```

## MainWindow.xaml.cs

```
using System.Windows;
namespace TestWpf {
  public partial class MainWindow {
    public MainWindow() { InitializeComponent(); }

    private void button_Click(object sender, RoutedEventArgs e) {
      MessageBox.Show("Hello World!");
    }
  }
}
```

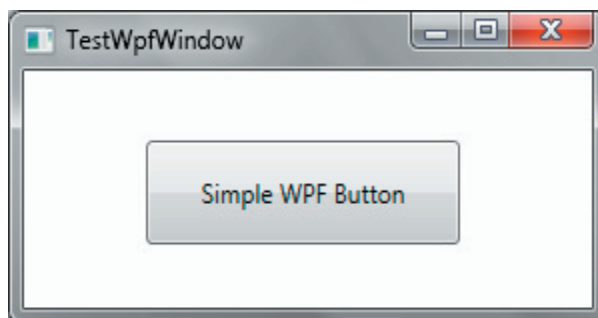


Рисунок 1. Результат выполнения WPF-приложения

контента для Web-страниц, не имеющее аналогов как по удобству разработки, так и предоставляемой функциональности.

Отличительной особенностью данной технологии является то, что разработка кода приложения происходит на языке программирования C# 3.0. Сама же среда исполнения является частью платформы .NET 3.5 Service Pack 1.

Среда включает в себя подмножество BCL (Base Classes Library – Библиотека Базовых Классов), расширенную специально с учётом характерных особенностей платформы Silverlight 2.0. В библиотеке классов также присутствуют многие из стандартных элементов управления WPF, и, разумеется, есть возможность расширять существующие и создавать новые.

Таким образом, для программистов на C# 3.0, работавших в WPF, не составит труда разрабатывать приложения для данной платформы, и даже портировать на неё Windows приложения написанные для платформы .NET 3.x с использованием WPF.

Также данная технология поддерживает взаимодействие с Веб-сервисами, разработанными с использованием технологии WCF и не только и со страницей в которую оно встроено через её DOM модель.

Пример Silverlight 2.0 приложения (см. пример и рис.2):

Как видно из приведённого примера, отличия Silverlight 2.0 приложения от WPF приложения - минимальны

## TestPage.html

```

<html>
  <head>
    <title>TestSilverlight</title>
  </head>
  <body>
    <br />
    <div id="silverlightControlHost" align="center">
      <object data="data:application/x-silverlight,"
        type="application/x-silverlight-2"
        width="300" height="145">
        <param name="source" value="TestSilverlight.xap"/>
        <param name="background" value="white" />
        <param name="minRuntimeVersion" value="2.0.31005.0" />
        <param name="autoUpgrade" value="true" />
      </object>
    </div>
  </body>
</html>

```

## App.xaml

```

<Application
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  x:Class="TestSilverlight.App" />

```

## App.xaml.cs

```

using System.Windows;
namespace TestSilverlight {
  public partial class App {
    public App() {
      this.Startup += this.Application_Startup;
      InitializeComponent();
    }

    private void Application_Startup(object sender, StartupEventArgs e) {
      this.RootVisual = new Page();
    }
  }
}

```

## Page.xaml

```

<UserControl x:Class="TestSilverlight.Page"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Width="300" Height="145">
  <Canvas Background="Silver">
    <Button Canvas.Left="70" Canvas.Top="45"
      Height="48" Width="157"
      Click="button_Click"
      Content="Simple Silverlight Button" />
  </Canvas>
</UserControl>

```

## Page.xaml.cs

```

using System.Windows;
using System.Windows.Browser;

namespace TestSilverlight {
  public partial class Page {
    public Page() { InitializeComponent(); }
  }
}

```

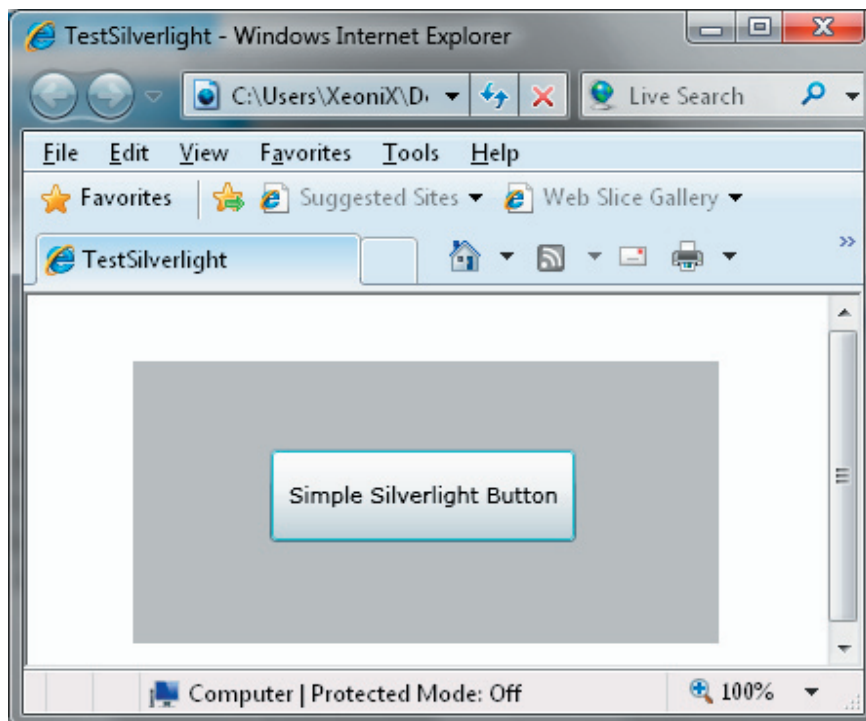


Рисунок 2. Результат выполнения приложения Silverlight 2.0

---

## 5. Выводы

---

Технологии WPF и Silverlight 2.0 ещё не получили широкого распространения, ввиду своей новизны, однако у них есть большие перспективы. На сегодняшний день является актуальным использовать Silverlight 2.0 в крупных проектах по разработке Web-базированных информационных систем ввиду удобства

разработки и возможностей, предоставляемых для разработчиков, чтобы удовлетворять постоянно растущим требованиям пользователей.

---

### Литература

1. The Official Microsoft Silverlight Site [www.silverlight.net](http://www.silverlight.net)
2. Silverlight дизайн - [www.silverlight.ru](http://www.silverlight.ru)