

Role of .NET in Defining, Executing, and Managing Workflows

Naeem Noor Ali Charolia¹, Muhammad Nadeem²

¹MSCS SZABIST, Karachi

²SZABIST,

Karachi, Pakistan

nadeem@szabist.edu.pk

Naeem Noor Ali Charolia

Abstract: In the current era of software development where most of the projects are based on workflows (i.e. Document Management, Loan Approval), developing workflows is a very lengthy time consuming process without any visual appearance.

Therefore it would be a challenging task for a software engineer to develop such workflow without any visualization. Microsoft has introduced in .NET framework 3.5, an extension which facilitates the developer to process easily on object oriented data model.

In this report, I have tried to describe the need of workflow, how it can help us, and how Microsoft helps us in creating and managing these workflows?

This report starts with an introduction of workflows, how it executes, and then later discusses WF, problems catered by WF, how WF works with Document Approval. In the end some future direction and conclusion is also provided.

Keywords: WF, Windows Workflow Foundation.

1. INTRODUCTION

1.1 What is Workflow?

In early 1980s, David Siegel used a term “Workflow” for his software industry, which deals in business process automation [1].

An essential way of documenting the activities involved in completing a task is Workflow. Each activity can be performed by either machine or people. It can be simple like writing the sequence of page or can be complex like managing documents that are shared among number of people for editing.

Windows Workflow Foundation provides a general purpose programming framework through which business rules and control flows are implemented graphically.

A workflow is sequence of steps needed to complete a specific task on the basis of certain decision and rules. For example while ordering pizza at Pizza Huts, we tell cashier the type of pizza. The cashier gives this information regarding the order to the cook. The cook gathers all the ingredients as per order and puts this ingredient in the oven. When pizza is cooked, cook gives the pizza to the cashier. A counter person collects the payment and completes the workflow [2].

1.2 Why Do We Need a Visually Managed Workflow?

- i) **Spaghetti Code:** As a developer, without workflow system, one has to write long spaghetti code. Workflow system allows to get rid of the long spaghetti code and design the business process in a visual manner.
- ii) **Easy to Understand:** It is very hard for business expert to understand the code of the workflow. Visual designer help them to understand and manage the workflow.
- iii) **Easy to Maintain:** Visual designer helps to maintenance/modify to workflow easily.

1.3 Benefits of Using Workflow

- i) **Easy to Understand**

Workflow provides the standard way in order to understand the flow behind the process. As we know understanding visual diagram is easier than understanding the code. Moreover, the business experts (who are non-technical) can easily understand the process defined in the workflow.

- ii) **Writing Long Program**

Developing workflows without visual designer mean that you have write long code as well as tracking the bug would be more problematic. With visual interface most of the workflow provides designer which generates irrelevant code irrespective of business flow like dropping an activity will generate Activity class in the code.

iii) **Time Efficient Development**

As far as development time is concerned, writing code for manual process takes long time rather than implement a business process on workflow.

iv) **Declarative Model**

Workflow allows explicitly / declaratively model to design the control flow of application rather than writing business logic in the code. As a result we can track the execution, inspect the logic, and change it at runtime.

v) **Manage Audit Trail**

Modern workflows provide classes (Application Programming Interface) to monitor the state of the workflow. Such classes help to manage the audit trail.

vi) **Information to Relevant Person**

There can be different type of stakeholders while implementing the project. It is hard to implement anything which is not understandable to all stakeholders. The workflow diagram provides information in such a way that it can be easily understandable to every stakeholder.

vii) **Easy to Monitor the State**

Sometimes we need to monitor the state of the workflow. Workflow also provides APIs through which we can monitor the state.

1.4 Role of .Net in Managing Workflows

Microsoft .NET 3.0 was released in November 2006 but launched officially in January 2007 with Windows Vista – along with Windows Communication Foundation (WCF) and Windows Presentation Foundation (WPF). This made the Microsoft’s standalone workflow technology platform which was designed to be hosted as part of any Windows application. The platform comprises of following:

- Workflow designer
- Library of workflow activities
- Programming model for hosting workflows in .NET applications
- Rules engine

- Workflow execution engine

Core features are based on built-in support for state management, workflow tracking, compensating transactions, exception handling and dynamic workflow updates.

.NET 3.5 launched with Visual Studio 2008 went a step further and introduced additional WF features primarily for better integration with WCF services including:

- i) New tools and features make it easy for developers to do the following.
- ii) Expose a workflow as a WCF service.
- iii) Receive messages to workflows via WCF service operations.
- iv) Send messages to WCF services from a workflow.
- v) Create durable WCF services that can survive a long running process by leveraging workflow state management

Productivity tools and workflow-related templates were also integrated with Visual Studio 2008. Also included were development enhancements to the rules engine to support early binding, operator overloading, and extension methods.

2. WINDOWS WORKFLOW FOUNDATION:

Microsoft .NET Framework 3.0 comprises of four technologies, which includes Windows Presentation Foundation (WPF), Windows Communication Foundation (WCF), Windows CardSpace and Windows Workflow Foundation (WF).

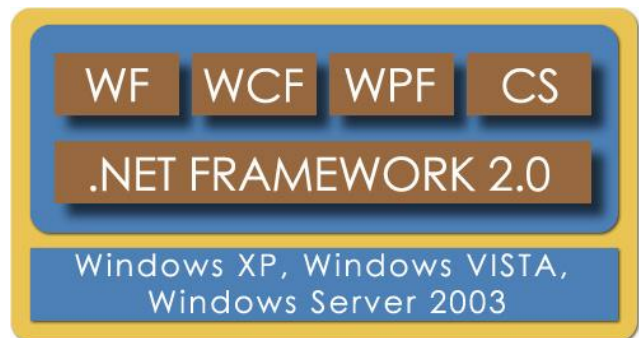


Figure 1: .NET Framework 3.0

Workflow Foundation (WF) is one of the key feature introduced in .NET Framework 3.0. An interesting aspect of Workflow Foundation (WF) is that the business process development is completely separate from actual implementation as Service Oriented Architecture (SOA) requires. The idea is to manage business logic in workflow and actual underlying implementation in any .NET compatible languages (like C#, Visual Basic .NET etc).

Note: This is an extension of Windows Workflow Foundation in .NET Framework 2.0.

2.1 Windows Workflow Benefits

- Windows Workflow Foundation (WF) help to create and understand business processes easily because it provides higher level of abstraction and visual representation for both (business domain experts and developers).
- Sometimes it is very hard to incorporate the changes in workflow (like changes in flow or business rules). By using the WF, it is easy very to incorporate the change, sometime without recompiling.
- Programming in Windows Workflow Foundation persuades developers to create a set of core activities that can be reuse in different applications.
- Comparing with UML, it is very much similar to Activity Diagrams.
- Its runtime gives a scalable and robust environment in which a workflow executes. Heavy processing workflows can be persisted to database when they become idle and reactivated when an external event occurs.

2.2 WF – Types of Workflow

A workflow is a movement of tasks through a process with different type of actors playing different roles. WF provides visual design and algorithms to create workflows for Windows application as well as any system using the embedded SOAP. By using Windows Workflow Foundation, we can create workflow's activities that make up a workflow.

A workflow contains set of activities. Each activity contains action that the flow occupies. An activity can be described as straightforward action or composite action that consists of several activities. Following are two types of workflows in WF:

2.2.1 Sequential Workflow

In this type of workflow, actions are executed in predefined order from start to end. For example: software installation – in which sequence of steps are required to complete the installation.

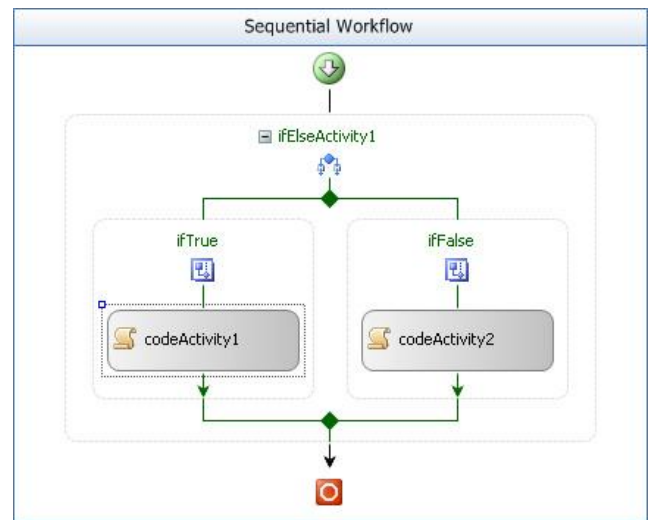


Figure 2: Sequential Workflow

2.2.2 State Machines Workflow

In state machine workflow, it contains set of states and transitions between states thus does not execute on predefined path. For example: E-Commerce website where you may need different types of states depending on user details. You may need to pay through credit card or PO for the transaction depending on the users details.

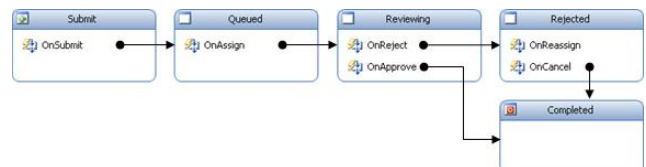


Figure 3: State Machine Workflow

2.3 Architecture of Windows Workflow Foundation

WF provides the classes, methods, and properties to program against the WF engine. Through this rich library, we can execute, stop and query a workflow. Controlling the state and modifying a workflow at runtime also possible through the library.

WF contains the following key building blocks:

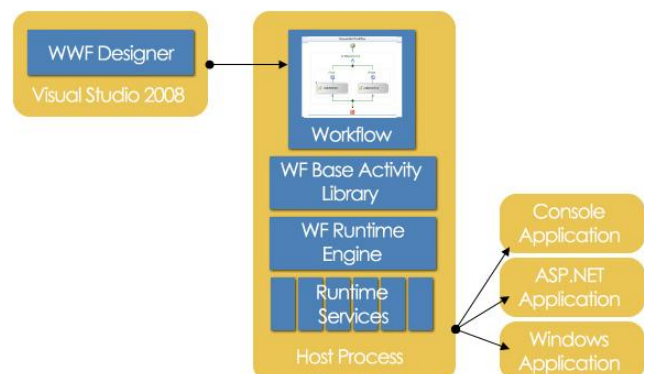


Figure 4: Architecture diagram

i) **Base Activity Library**

It provides base classes for custom activities, library to communicate with services and applications, control flow, event handling, conditions and looping, and state management. This library also provides container for other activities. For example exception handlers, compensation, and event handlers host every kind of activity.

ii) **Runtime Engine**

It provides workflow execution and state management.

iii) **Runtime Services**

It provides hosting facility for WF workflows.

In Windows Workflow Foundation, a workflow runs within a host process hosted on .NET application. A hosting application is an application that executes and manages its lifetime is a host. Any application that can execute .NET managed code can also execute workflow and host the workflow.

2.4 Benefits of Using WF

- i) **Integration with other Tools:** Windows Workflow Foundation can easily integrate with other tools like Microsoft Biz Talk Server.
- ii) **Easy to Understand:** Visual designer of WF is good enough that any stakeholder can easily understand the flow of activity at a glance.
- iii) **Efficient Time Development:** Visual Studio 2008 and Windows Workflow Foundation provide rapid development in developing any kind of workflow because designing a workflow in a visual environment is very time efficient.
- iv) **Declarative Model:** WF allows declaratively model to design a workflow which is very useful in terms of understanding business flow.
- v) **Manage Audit Trail:** Windows Foundation not only provides classes through which a workflow can be design but it also provides classes through which we can monitor the audit trail of the workflow.
- vi) **Information to Relevant Person:** It provides visual integrity through which every stakeholder can understand the workflow as well as they can get relevant information very easily.

- vii) **Easy to Monitor the State:** WF provides classes through we can monitor the state of the workflow.

3. DOCUMENT APPROVAL WORKFLOW USING WF:

3.1 SCENARIO

Here is an example of document approval system in which there are a list of documents and reviewers. The reviewer takes a document from the list and reviews the document.

After reviewing the document, if reviewer satisfied with the content then he/she can “Mark as Approved”, otherwise if the reviewer is not satisfied with the content of the document then he/she can “Mark as Rejected” the document.

3.2 Workflow Diagram

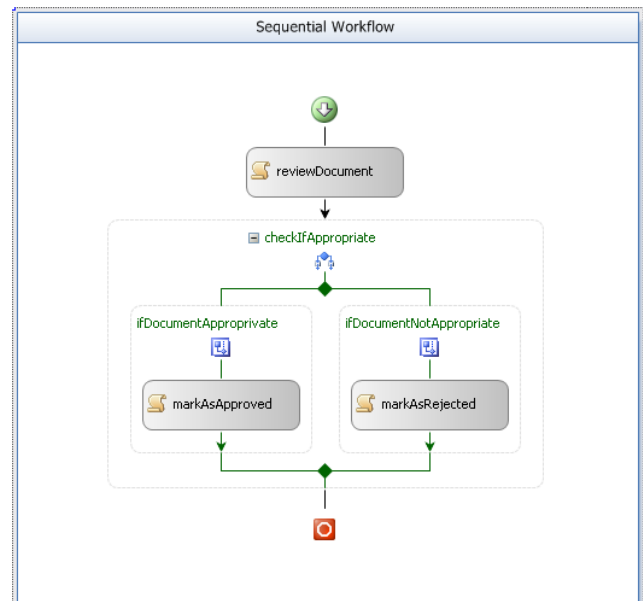


Figure 5: Document Approval (Sequential Diagram)

3.3 Code Snippet

```

public sealed partial class
    WFDocumentApproval:
        SequentialWorkflowActivity
    {
        char review;

        public bool IsApproved
        {
            get { return review == 'Y'; }
        }

        public WFDocumentApproval()
    }
  
```

```

{
    InitializeComponent();
}

private void
reviewDocument_ExecuteCode
(object sender, EventArgs e)
{
    Console.WriteLine("\n\nIs the document
        appropriate: ");
    review = (char)Console.Read();
}

Private void
    markAsApproved_ExecuteCode
(object sender, EventArgs e)
{
    Console.WriteLine("\n\n\n\n\n\t
        \tAfter review the document,
            it is found as approved.\n\n");
}

Private void markAsRejected_ExecuteCode
(object sender, EventArgs e)
{
    Console.WriteLine("\n\n\n\n\t\tAfter
        review the document, it is found
        as Rejected.\n\n\n\n");
}
}

```

3.4 Running Example

It is console based application, implemented in Visual Studio 2008 using .NET Framework 3.0. In this example I assume that a reviewer can take a document from the list of the documents which has to be reviewed. After reviewing the document, reviewer can approve or reject the document. Following are scenarios of both the cases:

3.4.1 Approve Case

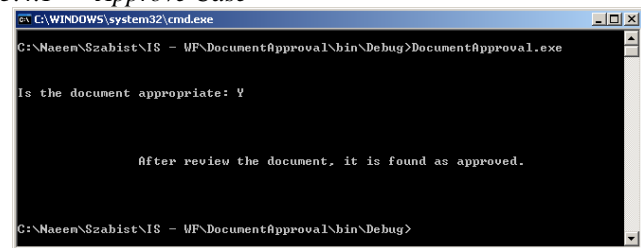


Figure 6: Approve Case

Let take an example for document approval. After reviewing the document system will ask for the content quality. If the reviewer satisfies with the content of the document, he/she marks as approved document.

3.4.2 Reject Case

3.4.3

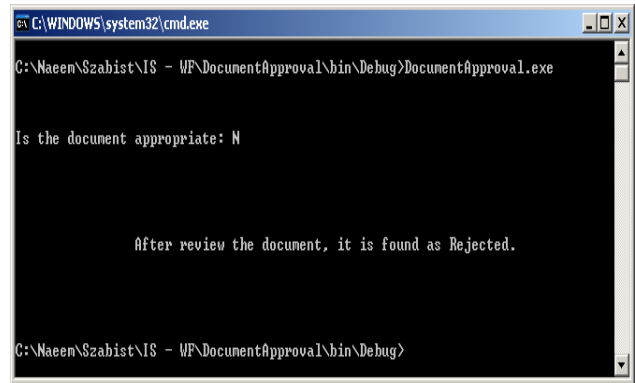


Figure 7: Reject Case

Now take an example for document rejection. After reviewing the document system will ask for the content quality. If the reviewer is not satisfied with the content of the document, he/she mark as reject document.

4. LIMITATIONS

4.1 Hiding Condition Criteria

Diagram show visual impact on process but it hides the criteria on which an activity take a decision to go right / left.

4.2 Extension of Classes

There are few classes like 'PolicyActivity' (which is a collection of Rule class), in which a developer might required to add a property in a particular domain. If a developer wants to change by inheriting a class, it is not allowed as such classes are sealed which is not inheritable [5].

5. FUTURE DIRECTION

5.1 Integration with Other Middleware

Windows Workflow Foundation allows communicating with Microsoft SharePoint Server 2003 and Content Management Server 2002 very easily. Microsoft has currently working on integrating with other tools/server like Microsoft FrontPage etc. [4].

5.2 Integration with Microsoft Project

As a management perspective, Microsoft Project is an excellent tool to manage the task flow and its estimation. Integration WF using would be more convenient for the stakeholder from project manager to developer in a software project teams etc. In this regard Microsoft is introducing a new application with the name of "Microsoft Project Server", which will help the community.

6. CONCLUSION

With the strong base, a building will never stable for long time. Same is the case of workflow, WF provide string foundation that caters the needs to design a business process, development challenges like investment of time and effort, and support the energetic behavior that people require.

Windows Workflow Foundation is a powerful tool which reduces the complexity of the code, supports all stake holders that involves in the development of the whole business process, and increase the productivity.

(2007) The IEEE website. [Online]. Available: <http://www.ieee.org/>

REFERENCES

- [1](2009) The Wikipedia website [Online]. Available : <http://en.wikipedia.org/wiki/Workflow>.
- [2] (2009) The White Paper on Microsoft Windows Workflows [Online]. Available: [http://www.global-infonet.com/Libraries/CustomDocLibrary/Windows%20Workflow%20Foundation\(WWF\).sflbdownload=true](http://www.global-infonet.com/Libraries/CustomDocLibrary/Windows%20Workflow%20Foundation(WWF).sflbdownload=true).
- [3] Hajo A. Reijers. Design and Control of Workflow Processes. New York: Springer-Verlag Berlin Heidelberg, 2003.
- [4](2009) Razi Bin Rais. "Understanding the Workflow Foundation (WF) from a Business User's Perspective." [Online]. Available : Internet: <http://www.codeproject.com/KB/dotnet/UnderstandWWF.aspx>
- [5] (2009) Ahmed Nagy. "Windows Workflow Foundation design and Somelimitations[Online]. Available:<http://forums.microsoft.com/MSDN/howPost.aspx?PostID=1331914&SiteID=1>.
- [6]Dimitrios Georgakopoulos, Mark F. Hornick, Amit P. Sheth. "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. Distributed and Parallel." V3, no. 2, pg: 119–153. April 1995.
- [7] Charles Plesums. "Workflow Handbook." 2nd ed., 2003. Pearson. 2003.
- [8]K. Scott Allen. "Programming Windows Workflow Foundation – Practical WF Techniques and Examples using XAML and C#, 2006.
- [9]W. M. P. vander Aalst, A. H. M. ter Hofstede, "Workflow Patterns: On the Expressive Power of Workflow Languages." pg 1 – 20. Technical Report DAIMI, 2002.
- [10] (2009) The "Windows Workflow Foundation.[online]. Available: <http://www.microsoft.com/net/wfdetails.aspx>.