



FEED YOUR BRAIN[®]

Workflows and SharePoint

Going with the flow

Gustavo Velez

Preface by Eilene Hao, Program Manager SharePoint Workflow at Microsoft Corporation



Workflows and SharePoint: Going with the Flow



Gustavo Velez



WORKFLOWS AND SHAREPOINT: GOING WITH THE FLOW

© KRASIS CONSULTING S.L. 2009

ALL RIGHTS RESERVED. NO PART OF THIS BOOK MAY BE REPRODUCED, IN ANY FORM OR BY ANY MEANS, WITHOUT PERMISSION IN WRITING FROM THE PUBLISHER.

ISBN: 978-84-936696-0-7

PRINTED IN THE UNITED STATES OF AMERICA

Preface

We may not realize it, but we participate in workflows every day! From asking someone to approve an expense to collecting feedback on a proposal, the things we do are almost always part of a series of steps that need to happen to accomplish a goal. But with so much going on, it's sometimes hard to remember all the steps, who to contact, and what to do next. And for processes that involve a lot of people, it's hard to keep track of where they left off and in whose hands.

SharePoint workflows are such an exciting part of the platform because they strive to tackle all of these problems. They increase productivity by focusing users on what they need to do and automating busy work tasks like sending nag mails and archiving data. They streamline processes by applying rules and application logic to the data that people interact with, such as comparing safe limits before routing expense reports to an approver. By recording progress along the way, they also provide visibility into a process and keep processes from getting lost. A powerful tool for managing document-centric processes, these workflows help drive what SharePoint does best: human collaboration.

Of course, since all processes are different, that power would be nothing without a framework underneath it that allows us to write the custom logic and flow that we need. And so, SharePoint hosts Windows Workflow Foundation to get that flexibility and programmability. Writing workflows for SharePoint can be challenging, but hopefully, understanding of just a few key concepts and SharePoint workflow quirks will make the process a lot easier and unlock a wonderful world of possibilities. I hope that this book helps to demystify SharePoint workflows and assists you in your quest to make SharePoint processes come to life!

Eilene Hao
Program Manager SharePoint Workflow
Microsoft Corporation

Acknowledgements

When I saw the inaugural presentation of the Windows Workflow Foundation at PDC in September 2005 (Los Angeles, USA), my immediate impression was: this is the quintessential application for Microsoft servers. Since then, the Foundation - integrated exclusively in SharePoint - has confirmed its capabilities and also revealed its weak points. Outside SharePoint, although the Foundation is used extensively, it constantly needs to defend its position against other workflow frameworks: inside SharePoint, it has proved to be one of the most value-added functionalities.

SharePoint 2003 allowed for a type of workflow using Event Handlers and developers appreciated the technology, utilizing it fully. The technology took a leap forward with the Workflow Foundation and its implementation in SharePoint and the developing community embraced it, recognizing its supremacy and user-friendly qualities while, at the same time, uncovering its shortcomings.

Workflows and SharePoint: Going with the Flow is a 'hands-on' manual for advanced users and developers. The approach is 'no-nonsense'; based on clearly-defined examples it describes and demonstrates the possibilities and applications of workflows in SharePoint. The programming examples graduate from basics, generating configurations with Sequential and State Machine Workflows, to more complex themes using ASPX and InfoPath forms. An attempt has been made to identify possible errors and problems and to provide solutions or workarounds, and in situations where that is not possible, to point the reader in the best direction for answers.

The hope is that developers at all levels will be able to pull this handy book from their reference shelf and, with minimal effort, create a functional workflow.

In recognition and appreciation, I would like to acknowledge the people involved in making this project possible. The editorial assistance of Lorraine Weber was invaluable, and although her background is not technical, she has an open mind and the ability to understand complex technical issues; this is a book written by four hands, even though she didn't want to be mentioned as one of the authors. Juan Carlos Gonzalez Martin of the Centro de Innovación en Integración (Integration and Innovation Center, CIIN, in Cantabria, Spain) and Rolf Eleveld from Infusion Development in Dubai agreed to read the text and sift out errors and inconsistencies. And finally, Jose Manuel Alarcón, editor of Krasis, who ensured the book's publication.

The synergy of SharePoint and the Windows Workflow Foundation are here to stay: a new version of SharePoint that will enhance this relationship is in the wings; two powerful Microsoft products with a bright future.

Gustavo Velez
11 February 2009

Contents

PREFACE	v
ACKNOWLEDGEMENTS	vii
CONTENTS	ix

PART I: THE WINDOWS WORKFLOW FOUNDATION AND SHAREPOINT

1. Introduction, Workflows in SharePoint, Architecture	1
Workflow Foundation Architecture and Components	2
Sharepoint as Host for the Windows Workflow Foundation	5
Types of Flows	6
2. Workflows in SharePoint	11
Introduction and Default Workflows in WSS and MOSS	11
Workflow Configuration in Lists and Libraries	13
Workflow in Action	17

PART II: CREATION OF WORKFLOWS WITH SHAREPOINT DESIGNER 23 |

3. Creating Workflows with SharePoint Designer	25
Introduction	25
Interface and Components	26
Building a Workflow with SharePoint Designer	28
SharePoint Designer Workflow Case	28
Development Environment	30
Site Topology	30
Building the Workflow – Definition	32
Building the Workflow – Steps	32
Final Modifications – Forms	46
The User Experience	46
Bits and Pieces	48

**PART III:
CREATION OF WORKFLOWS
WITH VISUAL STUDIO 49**

4. Creating Sequential Workflows with Visual Studio 51
Introduction..... 51
Creation of a Sequential Workflow with Visual Studio..... 54
 Sequential Workflow Case..... 54
 Building the Workflow – Flow Design..... 55
 Building the Workflow – Flow Configuration..... 58
 Building the Workflow – Coding the Flow..... 69
 Building the Workflow – Final Notes..... 75

5. Installation and Configuration of Workflows 79
WORKFLOW DEPLOYMENT..... 79

6. Creating State Machine Workflows with Visual Studio .. 85
STATE MACHINE WORKFLOWS 85
Creation of a State Machine Workflow with Visual Studio..... 87
 State Machine Workflow Case..... 87
 Building the Workflow – Setting the States..... 88
 Building the Workflow – Setting the Transitions 93
 Building the Workflow – Activity Configuration..... 97
 Building the Workflow – Source Code 103

**PART IV:
FORMS FOR WORKFLOWS 109**

7. Creating ASPX Forms for Workflows 111
ASPX Forms Workflow Case 111
Site Topology..... 113
Comments about architecture 113
Building the Data Objects..... 114
Building the Workflow – Flow Design 123
Building the ASPX Pages – Association Form..... 131
Building the ASPX Pages – Initiation Form 142
Building the ASPX Pages – Modification Form..... 149
Building the ASPX Pages – Hermes Task Form..... 154
Building the ASPX Pages – Apollo Task Form 162
Building the Workflow – Flow Code..... 165
Final Comments..... 172

8. Creating InfoPath Forms for Workflows 175
InfoPath Forms Workflow Case..... 175
Site Topology..... 177

.....

Some Explanations about the Design of the Flow	178
Building the Custom Lists Webservice.....	178
Building the Workflow – Flow Design	180
Building the InfoPath Pages – Association Form.....	185
Building the InfoPath Pages – Initiation Form.....	197
Building the InfoPath Pages – Modification Form.....	199
Building the InfoPath Pages – Task Form.....	202

**PART V:
ACTIVITIES AND ANNOTATIONS225**

9. Creating Activities for Workflows227

Activities.....	227
Creating a Custom Simple Activity.....	228
Example Business Case for a Simple Activity	228
Identifier Generator Webservice.....	228
Creating a Custom Simple Activity.....	229
Using the Activity in SharePoint Designer.....	247
Creating Composite Custom Activities	253
Bits and Pieces	258

10. Bits and Pieces 261

SharePoint Workflow Object Model.....	261
WebServices for Workflows.....	264
Web Parts Pages Webservice.....	265
Workflow Webservice	270
Workflow Reportage.....	275
Bits and Pieces	282

PART I:



THE WINDOWS WORKFLOW FOUNDATION AND SHAREPOINT

Windows SharePoint Services (WSS) and Microsoft Office SharePoint Server (MOSS) are Microsoft tools that allow for collaboration and communication within organizations, enabling the smooth flow of documents and information. WSS is the core of SharePoint, providing the infrastructure for authorization and authentication, search and store of information and additional services to ensure validity and security. MOSS adds a functionality layer above WSS, creating the possibility to work with Business Intelligence, different types of servers (Excel, InfoPath) and more advanced search capabilities, among other tasks.

Part one gives an introduction to the theory and practical use of workflows in SharePoint: what is the Windows Workflow Foundation and its relation with SharePoint, the default implementation of flows in out-of-box Portals and the user-experience for configuration and utilization.

Introduction, workflows in SharePoint, architecture

Daily work routines involve a series of activities, decisions and rules that can be grouped in an organized way within a workflow. Workflows exist everywhere and the programming world is no exception. In a traditional programming pattern, the instructions that describe and execute a workflow are integrated into the logic of the program, making it cumbersome to maintain and reuse. Windows Workflow Foundation (WWF) is a Microsoft technology that provides the model, engine and tools to define, execute and manage workflows in a consistent and efficient manner. It is a constituent part of Microsoft .NET Framework 3.0 and 3.5, together with the Windows Communication Foundation (to build and run interconnected systems) and the Windows Presentation Foundation (for the presentation layer of Windows applications). WWF is not a stand-alone application, it has no user interface nor does it have an internal means to control the application directly. Windows Workflow Foundation is a software groundwork designed to allow the use and application of workflows within other software applications.

SharePoint is the first Microsoft Server employing the Workflow Foundation as a standard workflow engine. The previous SharePoint versions realized simple workflows using the Event Handlers framework of its Object Model. The integration between SharePoint and the Workflow Foundation is realized using SharePoint Lists and Libraries and every level of SharePoint is accessible. Applications include the authorization and automation of tasks, monitoring processes, automatic observation and enforcement of business rules, and principally, document-centric workflows; the cyclic procedures that a particular document undergoes, from reviewing to approval. The combination of Workflow Foundation and SharePoint provides a way to isolate and encapsulate the business logic related to flows from the SharePoint application logic, thus making the system more reliable, testable and easy to maintain. Another important feature is the ability to reuse code: this is possible because a workflow that is created generically may be used throughout the entire SharePoint installation. Often different workflows look and function in very similar ways, with a minimum of effort it is possible to take an existing workflow and adapt it to a comparable task.

The range of possibilities for innovation in SharePoint workflows is exciting. With specialized developers experimenting with components and applications for the developing process, the boundaries continue to expand. This introduces the potential for creating entire workflow ‘ecosystems’ within a developing team and giving them the autonomy to isolate and redefine workflow problems from routine SharePoint activities.

Windows Workflow Foundation is an independent component of .NET 3.0 and 3.5 frameworks, and thus integrates seamlessly with Visual Studio 2005 and 2008. Currently, the WWF is only available in SharePoint (WSS and MOSS) 2007, but all signs indicate it will become standard in Microsoft servers, thus opening the workflow framework to all Windows server developers. This should result in a growing community of developers and an increasingly rich source of available information.

Workflow Foundation architecture and Components

Workflows are constructed using two types of components: activities and the plumbing source code, which ensures activities function in a coordinated manner. Activities are the fundamental building blocks of each workflow and are derived from the base class ‘System.Workflow.ComponentModel.Activity’. WWF has a set of standard, out-of-box activities and SharePoint contributes various specialized activities to interact with the portal; it’s also possible to program custom or purpose-built activities. Each activity is a microcosm containing all the attributes of object-oriented components; they are encapsulated, have a communication interface composed of properties and methods and can be reused by means of inheritance. Activities can be ‘simple’, meaning they don’t contain other activities (‘CodeActivity’, ‘DelayActivity’, etc) or ‘composite’. The latter is a container for one or more activities and is derived directly from the ‘CompositeActivity’ class; in general, most out-of-box activities in WWF are composite as are the majority of custom or purpose-built activities. (See Chapter 9 for detailed information on the development and configuration of Composite Activities for SharePoint)

Rules are an integral part of certain activities and determine their function (equivalent to decision statements in programming language: if, while, etc). Rules are inherited from the ‘System.Workflow.Activities.Rules’ class. WWF has its own Rules Evaluation Engine and they are grouped in ‘RuleSets’ (available for the developer in the ‘PolicyActivity’ activity), meaning there are a number of multiple rules that are evaluated in sequence. Rules may be modified at runtime and are stored independent of the workflow. Visual Studio serializes rules in a distinct .rules file using the same name as the workflow.

Workflows are physically serialized in files within the file system and are saved in a .XOML file. XOML is an acronym for Extensible Object Markup Language and is a serialization format of the Windows Workflow Foundation. Under a microscope,

the .XOML file format is identical to the .XAML file (Extensible Application Markup Language), which is a declarative XML-based language created by Microsoft to initialize structured values and objects. XAML, and consequently XOML, are open- specification protocols, published and maintained by Microsoft. In principle, a workflow may be developed entirely using any text or XML editor providing the files follow the appropriate schema and are compiled manually using the command-line workflows Compiler (wfc.exe) from WWF; wfc.exe accepts a simple .XOML file or a list of files as parameter and compiles them to a .NET assembly. If the flow requires code-behind files (.cs or .vb), they are included in the compilation process. That said, the preferred implementation to create and compile workflows is with Visual Studio and its sophisticated Toolbox. If basic modifications to an existing workflow (changing an output string, for example) are necessary, it may be desirable to compile the .XOML file directly using wfc.exe or via an automatic or predefined compilation process. As is illustrated in Chapter 3, SharePoint Designer Workflows lack assemblies, the workflow is saved in a list and the SharePoint DataBase and the files are compiled when the flow is initiated; SharePoint maintains the compiled flow in memory to speed up future execution performance.

Workflow Foundation uses a layered architecture, as seen in the bird's eye view below:

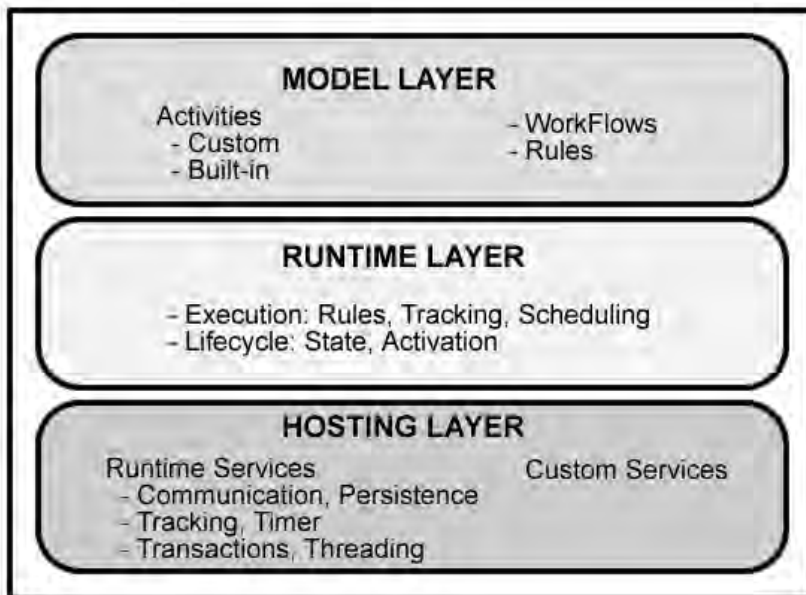


Figure 1 Windows Workflow Foundation

- **Model layer**, uppermost in the structure, is where developers build the code to run a workflow. This layer provides the out-of-box activities, the model for the construction of custom or purpose-built activities and the engine to construct rules, as well as the plumbing tools to configure and connect the activities. All the tools to build and operate workflows discharge their duty in this layer. Physically, the Model layer is organized in different 'NameSpaces' ('System.Workflow.Activities', System.Workflow.ComponentModel'). Additionally, this layer maintains the classes to communicate with Visual Studio and its design tools, the work horses of the developing experience familiar to all windows programmers
- **Runtime layer** ensures the execution aspects of the workflow and contains the mission-critical services required: for example, the state management service and the rules service that provide policy execution functionality are executing at this layer. The classes in the 'System.Workflow.Runtime' take care of the proper functioning of this layer and are implemented in a self-contained 'Runtime' engine (note that it is not a separate application but integrated in WWF); the methods in this class configure and control the workflow in all its phases. Although, the Runtime layer is the component that provides an execution environment for each workflow instance, the actual running of the flow needs a Hosting layer; the SharePoint engine in our case. Communication between the Runtime and Hosting layers occurs via the events that are thrown up by the Runtime and handled by the Host; each event carries an instance of the 'WorkflowInstance' class, which acts as a proxy to the actual instance managed by the Runtime engine and provides the necessary information to the Host
- **Hosting layer** is the connecting link between the Workflow Foundation and the outside world and supplies a package of services (Persistence, Timer, Tracking, Communication) needed to guarantee the control and management of the workflow. Although interfaces needed for this layer may be specially developed, WWF offers pre-built services with interfaces necessary to work with Framework 2 (and upwards) of .NET. And, of course, because SharePoint has built-in support for WWF, it is the Host *par excellence*. The Workflow Foundation supports the concept of external services in the form of Core/Local-Services, class instances that fulfill a defined purpose. The Core-Services defined by Microsoft are always present in the Foundation and the Local-Services (Custom/External-Services) are developed to serve any purpose. There are many Core-Services and they are indispensable for the proper functioning of a workflow instance; for example, the Persistence service saves the current state of the instance using a SQL DataBase when it is idle or in a suspended state. The Persistence service conserves server resources and allows long-running processes to be dehydrated and hydrated in an efficient way

SharePoint as Host for the Windows Workflow Foundation

As stated, Windows Workflow Foundation has no intrinsic execution processes; instead it is an in-process engine that runs within a host process. The host process provides the set of services needed for the lifecycle of the workflow. Any application that executes Windows Workflow Foundation and is able to manage their progression (start, stop, pause) is acceptable as host. Windows applications with the ability to execute managed code on the .NET Framework 2 (and upwards) can execute workflows and be a host. And, as every workflow requires a host to function, a WebService, WebPage, Windows Form Application, console application or Windows Service is welcome to host a workflow.

The host is in charge of initializing the workflow and controls its course of action, messaging and configuration-related tasks. To perform the task, WWF presents Namespaces, Methods and Properties in the Object Model to interact with the workflow: the 'WorkflowRuntime' class holds assorted Namespaces (Runtime, Hosting, Messaging and Configuration) that manage the timing and configuration of the flow.

Because SharePoint has an intrinsic relationship with the Workflow Foundation, it functions as host for the WWF run-time engine. SharePoint commands a number of Custom-Services that share tasks with other Core-Services and implement the necessary functionality:

- **Transactions** combine completed workflow steps into independent units to maintain data consistency
- **Persistence** enables the runtime engine to persist flows so they don't have to remain in memory until termination; this is for scalability reasons and to survive periods when the host process recycles or terminates. If the flow is idle, the state of the executing workflow is temporarily stored in the DataBase and removed from memory; at some point in the future, the flow is retrieved, reinstated and continues its execution
- **Scheduling** controls and manages the number of threads, their availability, priority and order of execution; it also provides a timing mechanism and prevents conflicts with the threads used by the SharePoint engine
- **Tracking** provides the capability to record performance and state information for monitoring
- **Messaging and Notification** offers a medium to communicate information to users and the external world

This format assures a seamless integration between SharePoint and WWF, but on the downside, some default Core-Services that work impeccably in other hosts have bugs in SharePoint: the Persistence Custom-Service of SharePoint doesn't work properly, for example. Furthermore, the standard WWF tracking provider supports third-party tracking providers but it is not sustained by SharePoint. SharePoint implements its own tracking service, but it is not as flexible as the original and it only implements certain reports and status indicators, extended reporting of workflows needs to be programmed manually.

Types of Flows

The Workflow Foundation offers two types of workflows: Sequential (derived from the 'System.Workflow.Activities.SequentialWorkflowActivity' class) and State Machine (derived from the 'System.Workflow.Activities.StateMachineWorkflowActivity' class). Workflows are able to work in two ways (or in combination) with (Sequential) system and with (State Machine) human interaction.

The traditional approach to a problem follows a set of pre-determined steps in a top-down process, the Sequential, the system-interaction workflow, mirrors this pattern in that actions transpire from a beginning situation and end after a predefined sequence of steps. Almost any undertaking could be described as a function of a Sequential Workflow; business authorization activities are a well-known type. Consider tasks such as order processing, expenses approval or purchase requests as typical scenarios of this kind of flow. Sequential Workflows are self-driven; once they are initiated they have a totally predictable behavior throughout the execution of the mission and they are determinist: the process can only go forward, it is impossible to go back. The control of the flow is defined with decision statements such as 'while' loops and 'if-else' branches.

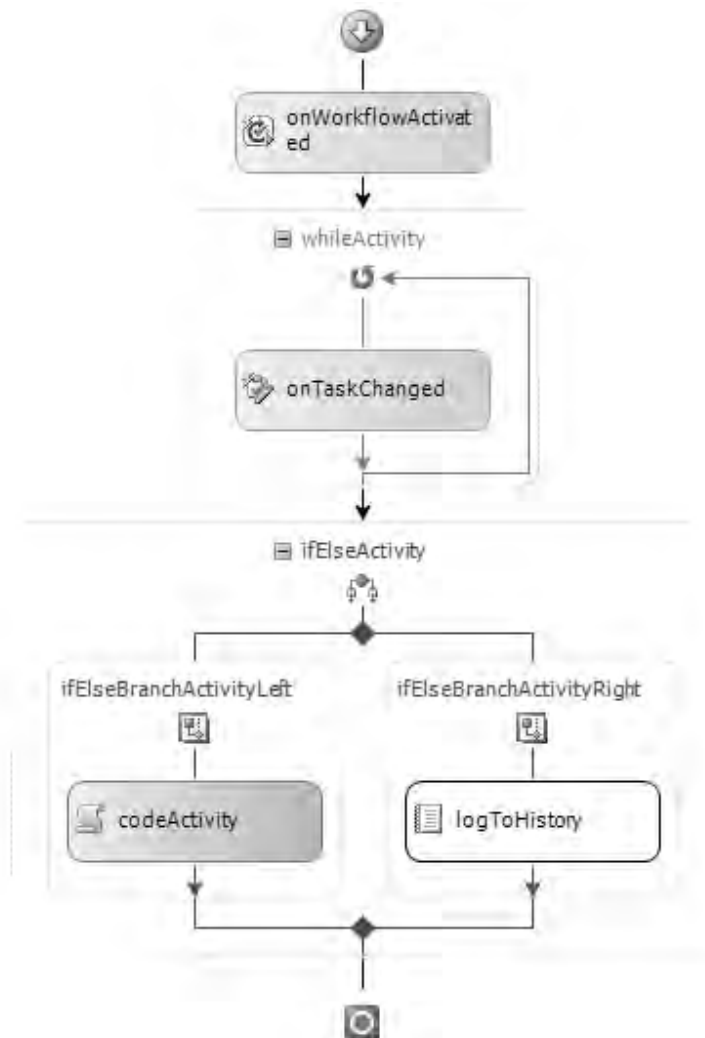


Figure 2 A Sequential Workflow Illustrated in Visual Studio Designer

Unlike programmable applications, real-time human processes are considerably less predictable. In the example of the order processing procedure, the final approver can return the order to the initial approver for corrections or to the requestor with a reject notice, that is to say, the flow may be reinitiated from any random point in the process. These flows are known as human-interaction or State Machine Workflows and they are very difficult to express using a clear Sequential Workflow because humans are in control, not machines. The State Machine Workflow does not have a defined path but represents a set of states and transitions between these states. State

Machine Workflows are event-driven: each state is activated after a predefined action occurs, the engine executes the steps needed and stops after completion of the next state: the transitions between states are triggered by external events that are raised by the host (SharePoint). There is no deterministic execution path between the steps because the workflow does not execute in a chronological order.

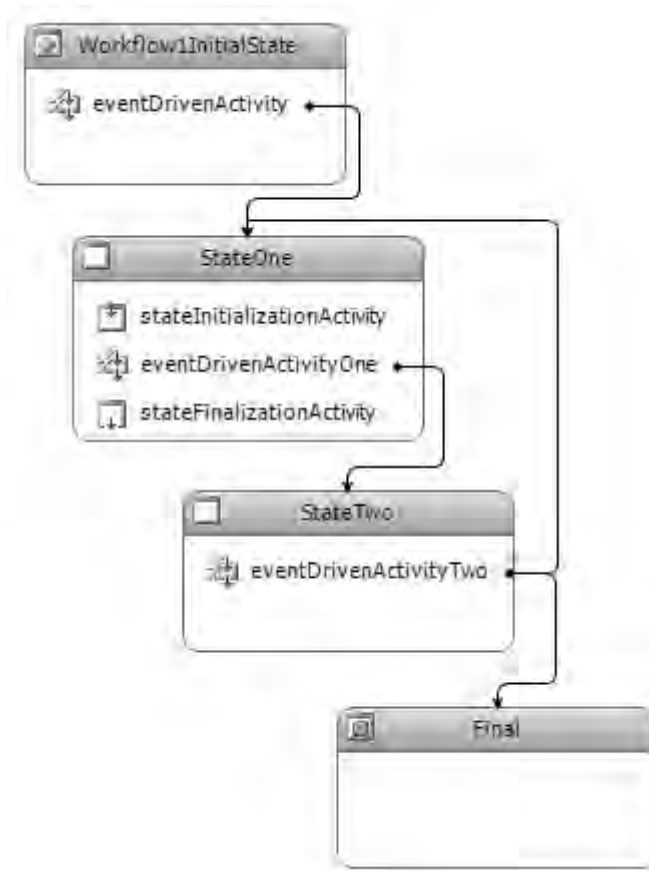


Figure 3 A State Machine Workflow displayed in Visual Studio Designer

Although the chosen methodology may not always be obvious; for example, when to use a State Machine over a Sequential Workflow and vice versa, it's fair to say that potentially any problem is resolvable with a Sequential or State Machine Workflow. State Machine Workflows offer a natural, rational model whereby the state and the present activity of the process being executed are more important than the order of the steps taken to complete it. In situations requiring many unplanned steps to track, and where it is impossible to represent all the potential routes to take, a State Machine Workflow is easier and more orderly to program.

Generally speaking the type of issues and the desired results dictate the choice of flows but sometimes, for processes that are difficult to model in one or the other, a combination may be implemented within a State Machine Workflow. In fact, as illustrated in Chapter 9, all actions used in State Machine Workflows are created using Sequential steps.

In conclusion, the primary difference between the two is that a Sequential Workflow is self-determined and therefore controls the direction taken; for State Machine flows, the host (through the user) sets the sequence and the course of action. It is virtually impossible to systematically recommend a preferred workflow type for any given situation; therefore it is critical to analyze the situation and consider the parameters before initiating a workflow. Be aware that an erroneous workflow choice will result in additional complications and often failure. In the course of the book, detailed information, practical experience and illustrative examples will assist the reader in learning to choose workflows wisely.



Gustavo Vélez
SharePoint MOSS MVP

Gustavo Vélez is Solutions Manager at Avansade, a global consultancy firm. He has many years experience developing Windows and Office applications, and several years of daily programming experience with SharePoint. The author's articles can be found in many of the leading trade magazines in English, Dutch and Spanish.

*He is also pleased to be Webmaster of www.gurd.net, the only Spanish-language site dedicated to SharePoint. Spanish-language readers may want to consult Vélez's book: *Programación con SharePoint 2007*.*

Vélez has been awarded MVP status (Most Valuable Professional) in the area of SharePoint MOSS.

All our authors have been awarded MVP (Most valuable Professional) status by Microsoft Corporation. In a yearly review covering more than 90 countries and technologies, Microsoft selects "the best and the brightest".

Workflows and SharePoint: Going with the Flow was written by a developer for developers; a comprehensive reference to guide users - step-by-step - through the process of creating a workflow. A case study allows the reader to follow the development of a flow, incorporating the myriad of possibilities and tools available to the developer.

The book's approach is 'no-nonsense'; based on clearly-defined examples it describes and demonstrates the possibilities and applications of workflows in SharePoint. The programming examples graduate from basics, generating configurations with Sequential and State Machine Workflows, to more complex themes using ASPX and InfoPath Forms. An attempt has been made to identify potential errors and problems and to provide solutions or workarounds, and in situations where that is not possible, to point the reader in the best direction for answers.

Workflows and SharePoint: Going with the Flow is divided into five areas for easy reference. The first two parts focus on information for the general reader; introducing the architecture and use of workflows followed by a description of the use of SharePoint Designer to create flows without programming. Part 3 provides information for the creation of Sequential and State Machine Workflows using Visual Studio and flows over into the next section covering the creation and development of Forms (Part 4). Finally, Part 5 reviews Activities, their use in Visual Studio and SharePoint Designer, and goes on to discuss other 'bits and pieces' related to programming workflows for SharePoint.

Workflows and SharePoint 2007: Going with the Flow is the kind of reference book that belongs on every developers bookshelf.

LEVEL - Intermediate to Advanced

