



Computer Science

---

**Harald Quist**

**Creating a Help Desk  
using SharePoint Workflow**

Bachelor's Project

C2009:02

# **Creating a Help Desk using SharePoint Workflow**

**Harald Quist**



This report is submitted in partial fulfillment of the requirements for the Bachelor's degree in Computer Science. All material in this report which is not my own work has been identified and no material is included for which a degree has previously been conferred.

---

Harald Quist

Approved 090603

---

Advisor: Andreas Kasser

---

Examiner: Martin Blom



## Abstract

Xeratech AB is a medium-sized company in Karlstad, Sweden. Part of their business involves support of their products. This support has been managed manually; incoming errands has been received, by phone or by e-mail, and afterwards sent to a consultant for processing. This approach lacks the ability to efficiently store these errands, to make useful reports based on the work done with it, and to automatically send out notifications and e-mail to support members involved with the errand.

The goal of this dissertation is to implement an errand support system (a help desk) able to do the above things automatically. Since Xeratech use mostly Microsoft products, and uses SharePoint as their intranet platform, a choice has been made to implement this help desk system as a SharePoint State Machine Workflow. A state machine workflow is a workflow consisting of states, transitions and events. This type of workflow has been chosen because of its resemblance to the life cycle of an errand: errands will, during its life time, change from one state to another in a non predetermined way. For instance, when an errand is created, it will start in the *New* state, the workflow will then, when a person has started working with it, transition to the *In Progress* state, followed by a number of states until finally its state is *Completed*, and the work with the errand is done.

This workflow will then be evaluated considering its ability to facilitate the implementation of the help desk system.



# Contents

<b>1</b>	<b>Introduction.....</b>	<b>1</b>
<b>2</b>	<b>Background.....</b>	<b>3</b>
2.1	Introduction.....	3
2.1.1	Purpose	
2.1.2	Goal	
2.1.3	Delimitations	
2.2	Specification description .....	5
2.2.1	Introduction	
2.2.2	Outline	
2.3	Microsoft SharePoint .....	6
2.3.1	Introduction	
2.3.2	Microsoft SharePoint Architecture	
2.3.3	WSS and MOSS	
2.4	Workflows .....	8
2.4.1	Introduction	
2.4.2	What is a workflow?	
2.4.3	When to use workflows	
2.4.4	Types of workflow	
2.4.5	Technology basics	
2.4.6	SharePoint Workflow	
2.5	Chapter summary.....	16
<b>3</b>	<b>Designing the help desk system.....</b>	<b>17</b>
3.1	Introduction.....	17
3.2	Other approaches .....	18
3.2.1	Application Automation	
3.2.2	Orchestrations	
3.2.3	Event Receivers	
3.3	Description.....	20
3.3.1	Terminology	
3.3.2	Actors	
3.3.3	Entities	
3.3.4	Help Desk State Machine Workflow	
3.3.5	Notifications (sub-workflows)	
3.3.6	Timers	
3.4	Chapter summary.....	32
<b>4</b>	<b>Implementing the help desk system.....</b>	<b>33</b>
4.1	Introduction.....	33
4.2	Setup .....	33
4.2.1	Microsoft Virtual PC	
4.2.2	Microsoft Visual Studio 2005	
4.2.3	Workflow extensions for Microsoft Visual Studio 2005	
4.2.4	Windows SharePoint Services 3.0	
4.2.5	Microsoft Office SharePoint Designer 2007	
4.2.6	Windows Server 2003	
4.2.7	Hardware	



4.3	Implementing in SharePoint .....	35
4.3.1	Site	
4.3.2	Custom Lists	
4.3.3	Form	
4.3.4	Tasks	
4.3.5	Permissions, Groups and Roles	
4.3.6	Fields	
4.4	Ticket Registering.....	43
4.4.1	Internal Registration	
4.4.2	External Registration	
4.5	The State Machine .....	46
4.5.1	State Machine Activities	
4.5.2	State Machine Events	
4.5.3	State Machine Classes	
4.5.4	Timer Jobs	
4.6	Administration .....	53
4.7	Chapter Summary .....	54
<b>5</b>	<b>Evaluation .....</b>	<b>55</b>
5.1	Functional evaluation.....	55
5.1.1	Benefits of using workflows	
5.2	Quantitative Evaluation .....	58
5.2.1	Testing	
5.2.2	Limits	
<b>6</b>	<b>Conclusion.....</b>	<b>61</b>
6.1	General Summary .....	61
6.2	State Machine Workflow conclusion.....	61
6.3	General Conclusion .....	62
6.4	Project Summary .....	63
6.5	Future work.....	64
	<b>References .....</b>	<b>65</b>
<b>A</b>	<b>Help Desk Site.....</b>	<b>67</b>
A.1	Team Site Template .....	67
A.2	Assigned To Rejected Notification Form .....	68
A.3	Delegated To Affirmation Form .....	68
A.4	Solved Verification Form .....	69
<b>B</b>	<b>Permissions and Groups .....</b>	<b>70</b>
B.1	Help Desk Permissions and Groups .....	70
B.2	Help Desk Permissions Levels .....	71
B.3	Incorrect Permissions.....	72

## List of Figures

Figure 1: The SharePoint core.....	7
Figure 2: A simple State Machine example. ....	11
Figure 3: WF's main components. ....	12
Figure 4: WSS interaction with other components. ....	13
Figure 5: The Workflow Designer for a sequential workflow. ....	15
Figure 6: The Workflow Designer for a state machine workflow. ....	15
Figure 7: Process automation and workflows in a perspective. ....	19
Figure 8: Overview of the entities in the Help Desk system.....	22
Figure 9: Overview of the states, transitions and events in the State Machine.....	27
Figure 10: Assigned To Affirmation -and Rejected Notification activity diagram. ....	30
Figure 11: Delegated To Affirmation activity diagram. ....	30
Figure 12: Solved Verification activity diagram.....	31
Figure 13: The Help Desk Site.....	36
Figure 14: The parts of a SharePoint List. ....	37
Figure 15: The Assigned To Verification Edit form.....	40
Figure 16: Register a new errand in the Client Support Site.....	45
Figure 17: Diagram of the states and activities in the State Machine. ....	46
Figure 18: Pseudo-code for the CheckFieldChanges() function. ....	48
Figure 19: The InProgressChangedActivity.....	50
Figure 20: Diagram of the classes in the State Machine. ....	51
Figure 21: Pseudo-code for the WorkflowItemsUpdateTimerJob.....	53

**List of tables**

Table 1: The forms in the Help Desk List and the fields that they contain..... 39

Table 2: The permission levels in the Help Desk system. .... 41

Table 3: The fields in the Help Desk List and their types. .... 42

Table 4: Limitations of SharePoint ..... 59

# 1 Introduction

The importance of a well functioning intranet is for many organizations a priority. Such an intranet should contribute with efficiency improvement of an organization by providing the components to do so. In the last few years, Microsoft SharePoint has become the choice of many organizations. Microsoft SharePoint provides a deep integration with many other Microsoft programs such as Office and Outlook. This integration facilitates, among other things, the sharing and organization of documents within an organization. It also provides workflows which can be used directly for initiation, tracking and reporting of day-to-day company activities such as document reviews- and approval.

Xeratech is a medium sized company in Karlstad using SharePoint as their intranet platform. They have a support (a help desk) where their clients can call with their errands. However, this support involves a lot of manual work, such as sending email to persons within the organization. This is neither efficient nor economic.

Since Xeratech works mainly with Microsoft products and use SharePoint as their intranet platform, a decision was made to make a SharePoint Workflow to reduce the amount of manual work needed in the help desk. This workflow should be able to facilitate the process of assigning an errand to a person, delegate the errand to another person, store information about the work with the errand so that reports and statistics can be made, and to automatically send out mails and notifications to the people involved in the work with the errand.

This workflow should divide the lifetime of an errand into different stages, such as *New*, *In Progress*, *On Hold* or *Complete*. Not only so that the help desk team members easily can see in what stage the errand resides, but also because the lifetime of an errand should not be predetermined, which means that an errand can move back and forth from different stages (such as between *In Progress* and *On Hold*) before completion.

Since the workflow is going to be handed over and managed entirely by Xeratech when the project time is over, the workflow also needs to be created in such manner so that its functionalities can be easily increased. It should also be a focus on using the SharePoint out-

of-the-box functionalities as much as possible, as this will facilitate the integration with the existing Xeratech intranet.

The dissertation is divided into six chapters:

- Chapter 1 contains an introduction to the dissertation.
- Chapter 2 introduces and describes the components needed to easier understand this dissertation, with a focus on describing workflows.
- Chapter 3 explains the design of the system. It also describes other approaches on how to implement a Help Desk, comparing with the chosen workflow approach.
- Chapter 4 contains an explanation of the implementation of the system.
- Chapter 5 contains both a functional- and a quantitative evaluation of the State Machine Workflow, but also a short evaluation of the other parts of the system.
- Chapter 6 presents the conclusions made.

## 2 Background

---

*This chapter will give an introduction to this dissertation and its main goals. It will also briefly describe the company Xeratech AB and their necessity of a new support handling system, as well as introduce some concepts to facilitate the comprehension of this dissertation, such as basic SharePoint architecture, workflows and the different types of workflows. The chapter will conclude with a short summary.*

---

### 2.1 Introduction

Xeratech AB is a company with offices in several towns in Sweden, offering document solutions and information- and transaction solutions connected to documents [1]. They are also an established service company and offer high quality support for their customers, and have a consulting team with specialist competence in a number of areas.

Xeratech is a company in expansion; from being a small-sized company in Karlstad to a company with seventy employees in several Swedish cities. The company consists of three departments; Print Management, Process Management and Supply Management. Today none of these have an adequate Help Desk system, making the support to customers more difficult.

Xeratech has now decided to make a help desk system to facilitate this process.

The main features of the system is that it will be able to register support errand by web or by phone (instead of just by phone as they have now), and then to be able to delegate this errand to an expert. The experts, and the coordinators of the support errands will be able to see and work with a list of errands, these errands will have different status, such as *New*, *In Progress* or *Completed* depending on how far in the process the errand has come.

The help desk system will have many possible statuses, and probably not every state will be used by all errands. Doing this with a sequential workflow would be cumbersome, as it would require more *while* loops and *if* branching [11] compared to a state machine. That is why the help desk system is a good example of a system that can be implemented using a state machine workflow, as a state machine workflow has no predetermined path and therefore is better at handling more complex workflow processes. Workflows are explained more in detail in chapter 2.4.

### **2.1.1 Purpose**

The purpose of this thesis is to plan and implement a Help Desk system that fulfills the specification (see 2.2) that has been set up by Xeratech, using a Workflow approach.

### **2.1.2 Goal**

Since Xeratech is a company using mainly Microsoft products, a natural result of this is that also the Help Desk system is to be implemented using Microsoft technology, and more concretely using the SharePoint platform (see 2.3.2). Also, the SharePoint platform has got extensive support for Workflow solutions.

Along with this comes the main question of this thesis:

- How can SharePoint Workflows facilitate the implementation of a help desk system?

This question will be answered in form of an evaluation of the proper implementation of the help desk system. The evaluation will be made considering different aspects of SharePoint Workflow.

### **2.1.3 Delimitations**

This work may very well extend to be a full-time job a long time for a person experienced in SharePoint technology and Help Desk applications. For that reason, I will have to delimit my thesis and implementation to the very basics of the Help Desk system, using as much as possible the out-of-the-box features that exists in the SharePoint, but without decreasing the functionality of the system. Also, even though the system comprise of other parts than just the workflow part, focus is still going to be at the workflow part, as this is the main subject of this thesis.

## 2.2 Specification description

### 2.2.1 Introduction

From the beginning, Xeratech had a fairly clear vision as to how they wanted the help desk system to look like in the final version. They also understood that, as this work is to be made in a certain time, and so the required functionalities are only the most important. The optional is to be made if there is some time left.

### 2.2.2 Outline

The following functionalities are requirements:

- To have both a telephone and an online function for receiving errands<sup>†</sup>.
- That the errands can be logged.
- That the system can provide enough information about the errands so that reports and statistics can be made. (Closely related to the previous requirement.)
- To have a support telephone where the client always has a telephone number where he or she can reach the Xeratech support team<sup>†</sup>.
- That a member of the support team can delegate an errand to another support member.
- That a client receives support based upon his or hers Service Level Agreement. This includes for instance timers that triggers when an errand has not been started with before the time established in the Agreement. (See more about Agreements in chapter 3.3.3.3).

Also, there is some functionality that is optional:

- A FAQ (Frequently-Asked-Question) database where clients can search for solved errands which are considered interesting by support team (interesting may mean that the errand has a solution that applies for various types of errands).

---

<sup>†</sup> Requirements that do not directly affect the implementation, but that still may needs to be taken in consideration.



- That the client should be able to see the status of his or her tickets from the client web site.

## 2.3 Microsoft SharePoint

### 2.3.1 Introduction

SharePoint products and technologies are server applications. They provide an integrated platform to plan, deploy, and to manage intranet, extranet, and Internet applications [5].

SharePoint as both a platform and as a product is a very extensive subject. It is also quite complex; it uses many components, such as the .NET Framework, C#, ADO.NET, and Microsoft Windows, as well as Server Applications such as IIS, Active Directory, SQL Server and Exchange Server [6]. For that reason, only the basic SharePoint Architecture will be described, plus a few features for the purpose of better understanding of the SharePoint platform. Most of the out-of-the-box features that will be used come from the SharePoint Workflows: those will be explained later in this chapter (2.4.6).

### 2.3.2 Microsoft SharePoint Architecture

At the very core, SharePoint itself can be summed up into three Microsoft technologies [6]:

- **SQL Server** – Since everything in SharePoint is based on tables stored in the database (except some standard files and XML definition files that must be stored locally ), the entire composition of application, site, site collection, page, documents and web parts are defined by data stored in the database.
- **.NET** – SharePoint uses the .NET Framework, and leverages all of the features and tools built into it. This includes the .NET 3.0 Windows Workflow Foundation.
- **IIS** – Enables SharePoint to manage its own infrastructure from a web application point of view, handling the creating of virtual directories and web sites within IIS.

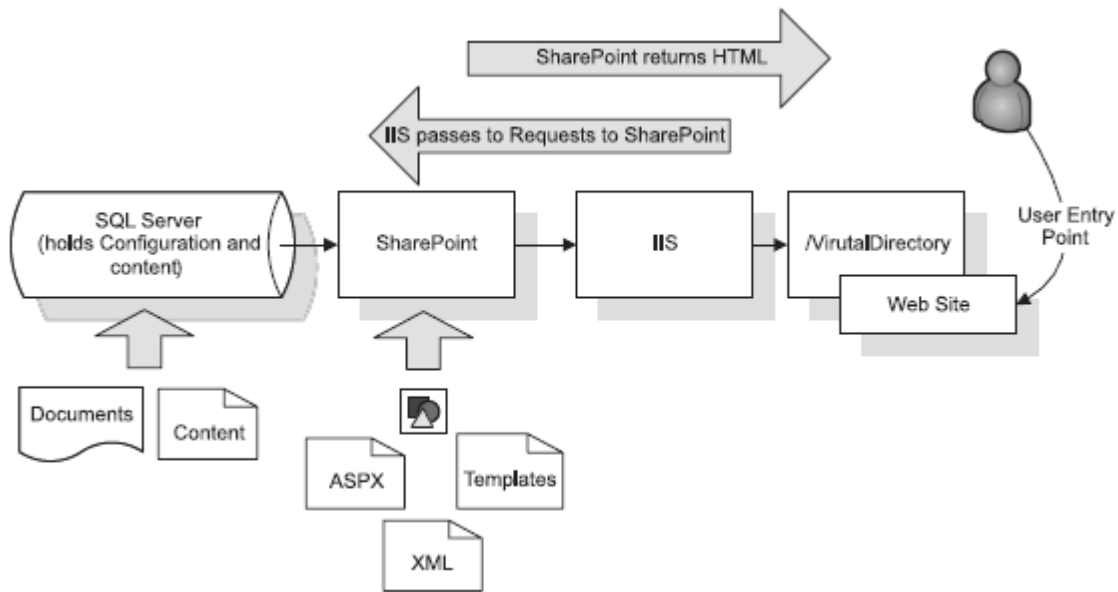


Figure 1: The SharePoint core.

As seen in figure 1, the SharePoint core is pretty straightforward. Worth mentioning is that the Web Site reside in a Virtual Directory (which means that the Web Site is accessed using the Virtual Directory instead of a physical folder name).

### 2.3.3 WSS and MOSS

SharePoint itself is comprised of two major parts: the Windows SharePoint Services (WSS), and Microsoft Office SharePoint Server (MOSS). It is important to distinguish these two and their features, as the first one is free and the second comes with an expensive license.

- **Windows SharePoint Services** - is a collection of services for Windows Server, and is the platform that provides the core SharePoint features and services. WSS manages all services on top of Internet Information Server (IIS). Some of the features are team sites, lists, libraries, workflow, and Office integration. WSS will be discussed more in chapter 2.4.5: Technology basics.
- **Microsoft Office SharePoint Server** - is an application built on top of WSS, providing additional services such as the user interface for the basic collaboration and publishing features, Portals, Enterprise Search and My Sites Services [7]. There are different MOSS editions, depending on the need of the buyer.

## 2.4 Workflows

### 2.4.1 Introduction

The remainder of this chapter will be an introduction to what workflow is, why they are used, and the different kinds of workflows. There will also be a brief introduction as to how SharePoint implements and facilitate the use of workflows. All this because the implementation, and the following chapters, will require a basic knowledge of workflows, and especially State Machine Workflows, which is the workflow used in the implementation of the Help Desk System (chapter 4).

### 2.4.2 What is a workflow?

A definition of the word workflow: “the process that defines and controls the completion of one or more tasks in order to bring about the realization of an identified goal” [10].

The key parts of this definition are “process”, “task” and “identified goal”. Breaking down the word one find *work*, which is the task, and *flow*, which is the process. Adding one more important aspect; the *goal* (the identified end result every workflow is targeted to achieve) to the definition of workflow, and the definition is clearer.

A more informal explanation is that workflow is how an item is moved from one person to another through a process, and the process defines the steps necessary to complete a piece of work [11].

Basically, there are two sides of the workflow coin:

1. The Machine-centric
2. The Human-centric.

#### 2.4.2.1 Machine-centric Workflow

In the Machine-centric, the computers are the primary participants and completers of tasks, and in the Human-centric, it's the people. Although there always will be some mixing of human versus machine participation in a workflow, they can still be classified based on who does *most* of the work. An example of a machine-centric workflow is assembly-line robotics

(for example, assembling cars). This is a workflow because the tasks which are to be completed must be done in a certain order in order to achieve a goal. There are numerous examples of similar workflows which have no human intervention simply because they are for instance too slow, the work is too dangerous, or for fraud-prevention reasons (like, for instance, in a Credit card approval workflow for online purchases).

#### **2.4.2.2 Human-centric Workflow**

Human-centric workflows doesn't work the same way; they generally need some sort of advanced reasoning, comparison, or abstract thinking that cannot be codified. Some sort of approval decision is also common for human-centric workflows; many human-centric workflows include a step where someone makes a judgment call on whether or not to proceed. The stereotypical example of a human-centric workflow is Document approval. In Document approval, no two documents are alike, and each of them requires a high level of abstract thinking and advanced reasoning to be approved.

The remainder of this paper will however only focus on human-centric workflows.

#### **2.4.3 When to use workflows**

It's worth mentioning when to use workflows, since workflows obviously aren't suitable in *every* application. The following list is a list of scenarios where workflows may come in hand [9].

- **Approval** – The need to get approval from one or more participants is a common aspect of business processes. The thing to be approved can vary widely, but in every case, some number of people must review the information. This could be for example revising the content, or appending comments, and then to indicate approval or rejection.
- **Coordinating group efforts** – Many processes require people to work in an organized way. By defining the steps of the process in an automated workflow, the group's work can be made more efficient.
- **Issue tracking** – A list of issues can be maintained by an automated workflow. The workflow may also assign issues to the appropriate people, and track the status of that resolution.

The choice of choosing workflows or not largely comes down to the answers of the following questions [4].

- Does any user interaction occur?
- Will the process run for a long time (more than a second or two)?
- Will the process need to pause to wait for another process to complete a task?
- Will the process be run many times (more than 25) concurrently?

If the answer is "Yes" for any of these questions, a workflow is a good choice.

#### **2.4.4 Types of workflow**

There are two basic types of Workflows, based on how the tasks are processed; Sequential Workflow and State Machine Workflow.

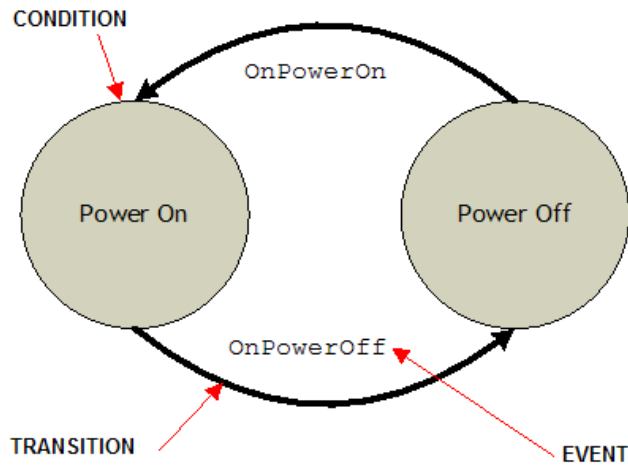
##### **2.4.4.1 Sequential workflow**

A Sequential Workflow is a workflow whose steps are performed sequentially (i.e. one after the other). A Sequential Workflow always follows a defined path, although it often includes loops, parallel branches and criteria-based branching. In a workflow, sequential means *continuous*, in contrast to the traditional concept of sequential in programming, where sequential means *without branching*.

##### **2.4.4.2 State Machine workflow**

A state machine is a model of behavior composed of a finite number of *states*, *transitions* between those states, and *actions* [12].

That is the usual definition of a state machine. In a workflow, state machines works in a similar way: the status, or situation, of the process being modeled is indicated by a *condition*, which is a set of circumstances, and a *transition* from one condition to another is caused by *events*. An example of this can be seen in figure 2.



*Figure 2: A simple State Machine example.*

State machines are fundamentally different from sequential workflow and often more complex. They are, however, much better at modeling complex human activities. Unlike sequential workflows, there is no predetermined path through the workflow. Instead, the path taken by the workflow is determined by the events that occur as the workflow is processing.

#### **2.4.5 Technology basics**

For the understanding of the human workflow support in Microsoft, two fundamental technologies require an introduction: Windows Workflow Foundation and Windows SharePoint Services. This section briefly describes each one of them.

##### **2.4.5.1 Windows Workflow Foundation**

The goal of Windows Workflow Foundation, WF, is to provide explicit support for creating applications that implements some kind of process, with multiple steps performed one after another in a defined order. An application built with WF consists of one or more workflows, each of which contains a number of activities. The activities of a workflow are executed one at a time by WF's runtime engine, with the execution order determined by the workflow itself.

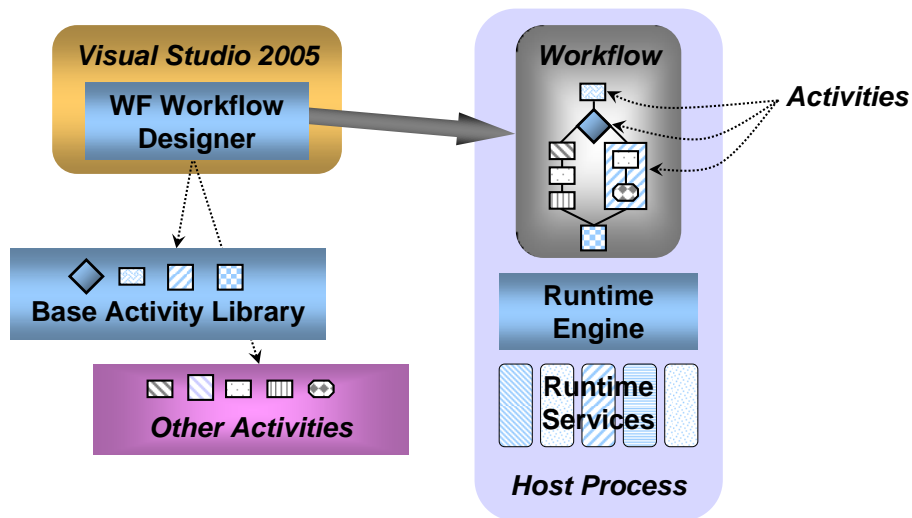


Figure 3: WF's main components.

A workflow, which is built from activities, is executed using the runtime engine. This execution depends on a set of runtime services that allows, for instance, tracking of its execution. This runs inside some host, which can be any Windows process (it can be everything from a simple desktop application to a scalable server). As indicated in figure 3, workflows can be created using WF Workflow Designer, used by Visual Studio 2005 (more about this in 2.4.6.3). An activity is just a class, so it's possible to create workflows purely in code (however, it can be very useful to use a graphical tool). WF provides a Base Activity Library (BAL), which includes a number of fundamental activities, including [9]:

- **IfElse** - executes the activities contained in two or more possible paths based on whether a condition is met.
- **While** - repeatedly executes one or more activities as long as a condition is true.
- **Sequence** - executes a group of activities one at a time in a defined order.
- **Code** - executes a defined chunk of code.

There's also possible to create own custom activities (depicted by the *Other Activities* box in picture 3).

#### 2.4.5.2 Windows SharePoint Services

WSS is a standard part of Windows Server 2003. WSS enables you to create sites, each of which contains document libraries and lists. The information in each site is stored in an SQL server, and ISS enables you to interact with the web. This can all be seen in figure 4.

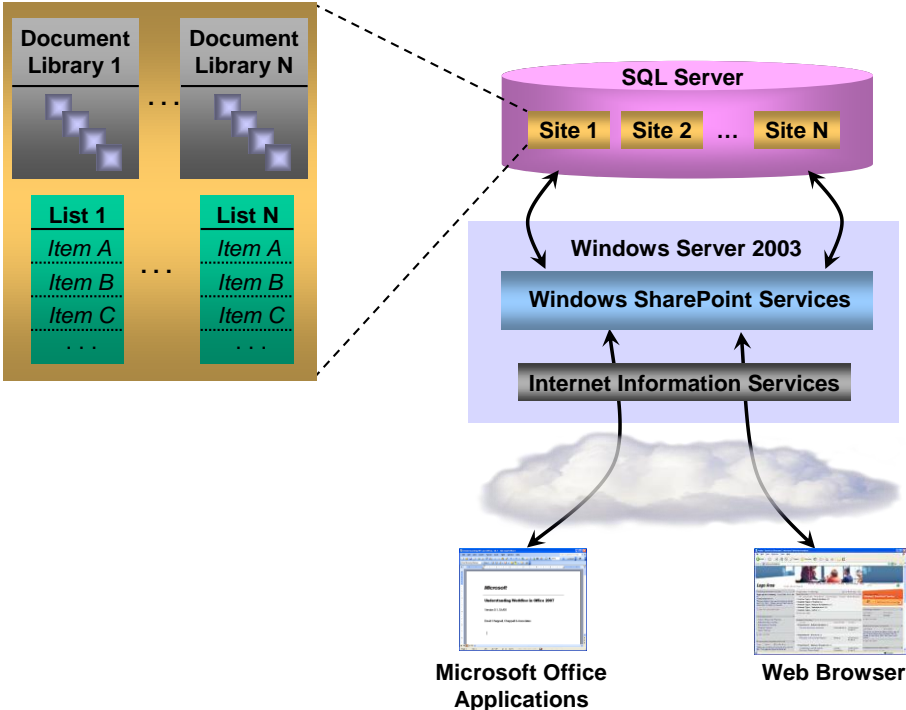


Figure 4: WSS interaction with other components.

## 2.4.6 SharePoint Workflow

### 2.4.6.1 Introduction

The SharePoint platform is focused on document-centric workflows: processes that a particular document needs to go through during its lifecycle. That process could be the reviewing, editing or approving the document for publication. There is also a focus on human-based workflow rather than just automating programmatic steps.

Below is a short introduction to the most important keywords, used in both SharePoint Workflow designer and in Visual Studio (Both of which will be explained later in this chapter).

- **Steps** – A workflow consists of one or more *steps*. Each step defines the *actions* and *conditions* that control the *activities* of that step. (Basically, one could see steps as a little more sophisticated *if* or *if/else* - clause).



- **Conditions** - The circumstances that signal that this step of the workflow should execute. Conditions determine at runtime whether an action for a step should execute. Example of conditions: *Compare Field*, *Title Field Contains Keyword*, and *The File Type Is a Specific Type*.
- **Actions** - An *action* is what happens during a workflow step. An action could be: Assign a To-Do Item, Create List Item, and Do Calculation.
- **Activities** – The building blocks of a workflow.

#### 2.4.6.2 Workflow in SharePoint Designer

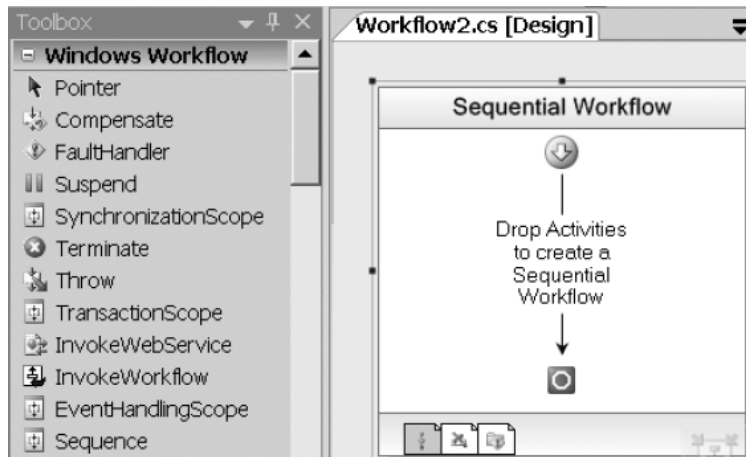
The main purpose of the SharePoint Workflow Designer, SPD, is to build new workflows from preexisting *activities*. Its primary audience is, according to Microsoft, non-IT department users [10]. This is because the tool lets you build a workflow simply without writing any code. Instead, Workflow Designer presents you with a wizard type interface in which you proceed through a number of steps to design and deploy your workflow. The tool is, for these reasons, fairly limited (for example, it has no support for state machine workflows).

#### 2.4.6.3 Workflow in Visual Studio

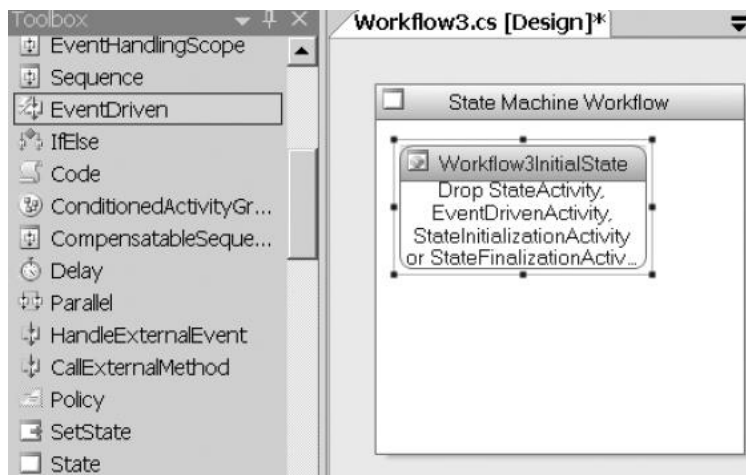
The SPD suffice for many situations, but when it comes to building a little more complex workflows, the Visual Studio 2005 Extensions for Windows Workflow Foundation is usually a better option.

Visual Studio has two built-in templates, the SharePoint Server Sequential Workflow Library and the SharePoint Server State Machine Workflow Library. Both Sequential and State Machine workflows are explained in section 2.4.4.

Both of these are built in a similar way: using the Workflow Designer, which is integrated in Visual Studio 2005. Figure 5 shows a sequential workflow made in the Visual Studio Workflow Designer canvas, and figure 6 shows a state machine workflow.



*Figure 5: The Workflow Designer for a sequential workflow.*



*Figure 6: The Workflow Designer for a state machine workflow.*

The Workflow Designer is pretty intuitive, and both the sequential and the state machine workflow are very similar to a sequence diagram and a state diagram, respectively, in UML notation. Although building a state machine is not quite as intuitive as the sequential workflow, the concepts and process are similar (at least at high level); they are both built using dragging-and-dropping activities from the toolbox to the canvas (see figure 5 or 6), the properties of the activities are set via the Visual Studio Properties window, and to perform the tasks of your workflow, both uses code-behind files where the code for the tasks is to be written.

## 2.5 Chapter summary

Although the architecture of SharePoint is fairly complex, the core architecture of SharePoint is not. It is comprised of three basic components: the *SQL Server*, which handles the database (where nearly everything in SharePoint is stored), the *.NET Framework* (along with the *Windows Workflow Foundation*) and the *IIS*, which handles and create the virtual directories and the web sites within it.

Two types of workflow exist in the SharePoint architecture: *sequential* and *state machine* workflow, where the latter is, although more complex, better at modeling human activities, as the path taken by the workflow is determined by the events that occur as the workflow is processing (i.e. there are no predetermined path in a state machine workflow). In SharePoint, every workflow is comprised of *steps*, *conditions*, *actions* and *activities*. The *steps* define the *actions* and *conditions* that control the *activities* of that step. The *action* is what happens during a workflow step, and the *activities* are the building blocks of a workflow.

## 3 Designing the help desk system

---

*This chapter contains the design of the help desk system. The chapter starts with a more thorough explanation of the system than in the introduction (see chapter 2.1). The different elements of the help desk system will be explained, as well as several diagrams on the architecture of the system. The different applications needed for the system to function (such as the Microsoft Virtual PC and the Microsoft Server) are presented, as well as a presentation of the actors of the system, and the entities of which the system is comprised.*

---

### 3.1 Introduction

The Help Desk system is comprised of several components. Basically, there are two different ways of interacting with the system; either external or internal. The external interaction is made by a client, either by using a simple web interface where he or she submits and registers his or her errand through a form, or by telephoning an internal actor that registers the errand. The internal interaction is made in the same interface no matter what role; the difference comes in permissions, where, for example the telephone operator have less permissions to make changes to an errand than an errand consultant, even though they are using the same interface. When an errand is registered, several parameters will be checked, such as the agreement between client and Xeratech in terms of support (this will be explained more in detail in section 3.3.3.3). Once an errand has been successfully registered and assigned to he or she whom will deal with the errand (i.e. the errand consultant, see “Actors” in section 3.3.2), the errand will be associated with a workflow. This workflow is the backbone of the Help Desk system; it sends notifications and verification tasks to involved actors based upon the particular agreement the client have, it keeps track of the time worked with the errand, and it stores the different actions made in a log, among other things.

## 3.2 Other approaches

Even though workflows was the direct choice as the approach for the Help Desk system implementation, there exists other ways, for example by using Application Automation [16].

### 3.2.1 Application Automation

The main purpose of both application automation tools and workflows is to automate business processes. The differences between them are not all clear; terms like “business process”, “task”, “job” etc. are used in both, but with different definitions. The automation terms are defined below. (The workflow terminologies are introduced in chapter 2.4.)

- **Job** – a single executable piece of work. A series of jobs make up an automated process.
- **Dependency** – criteria that determine the sequence in which jobs will execute.
- **Business process** – a series of jobs connected by dependencies.

Automation is generally defined as [17]:

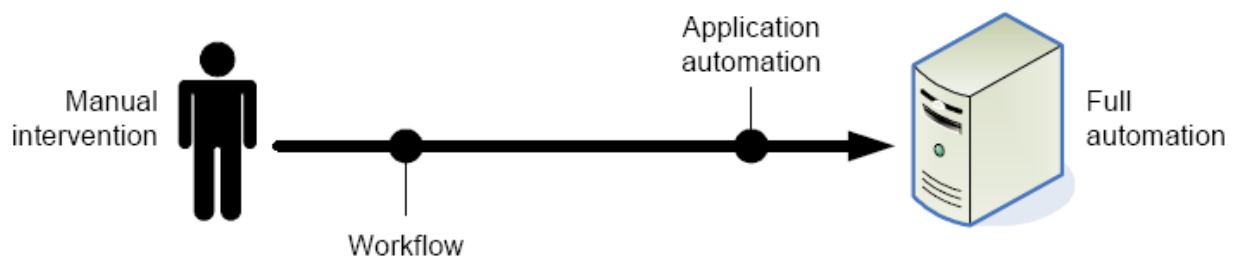
*The technique of making a process or a system operate automatically through the use of devices that take the place of human observation, effort, and decision.*

Application automations, or job schedulers as they are also called, facilitate automation of business processes and makes it possible to automatically launch business processes on specific days and times. True automation tools should make it possible to add dependencies (e.g.: only when Job A is completed, run Job B), and “if-then” logic that takes the place of an operator, checking the state of the system.

The difference between workflows and automation is that workflow is focused on managing work processes that consists of, at least partly, tasks performed by humans. Since these tasks cannot be fully automated (e.g. signing a document can never be made fully automated),

workflow tools focus on improving the overall work process by adding rules and notifications between tasks.

In contrast, application automation and its dependency logic is enforced without human intervention, and actively works to remove manual intervention. Figure 7 depicts a spectrum running from completely manual (left) to fully automated (right).



*Figure 7: Process automation and workflows in a perspective.*

### 3.2.2 Orchestrations

Orchestration describes the automated arrangement, coordination, and management of complex computer systems, middleware, and services [18]. An Orchestration is, in comparison with workflows, a larger and more complex interaction of resources and services within an enterprise and its trading partners. They are usually longer running than workflows, and they are usually more advanced. They generally need to run on a middleware product (such as BizTalk) [19].

### 3.2.3 Event Receivers

An event receiver is just what the name tells us: a receiver of events. That is, when an event is triggered, an event receiver receives the event and does something with it. In some specific cases, workflows could be substituted for simple event receivers successfully. However, some potential problems come along with this approach.

Workflows and event receivers have many similarities, including [4]:

- Each can be triggered when content is added or changed.
- Each can run a series of steps to complete some functional process.

Event receivers have several shortcomings compared to workflows when it comes to running long processes, and especially if the process often needs to be paused and stored. In these cases, workflows are preferable.

### 3.3 Description

#### 3.3.1 Terminology

- **Ticket** - A ticket is a file contained within an issue tracking system which contains information about support interventions made by technical support staff on behalf of an end user (client) who has reported an incident. The ticket will have a unique reference number, also known as a case, issue or call log number which is used to allow the user or support staff to quickly locate, add to or communicate the status of the users issue or request [21]. For simplicity's sake, in this Help Desk system, the terms ticket and an errand will be used synonymously.
- **SLA** - A Service Level Agreement, or SLA, is a part of a service contract where the level of service is formally defined. In practice, the term *SLA* is sometimes used to refer to the contracted delivery time (of the service) or performance [22]. In the Help Desk system, the SLA will have an active role, determining for instance the timeout for some *Notifications* (see section 3.3.5).

#### 3.3.2 Actors

Before explaining the different components (entities) of the system, it's appropriate to introduce the different actors that interact with the system. As mentioned in the introduction in section 3.1, there are two kinds of actors: internal and external actors.

##### 3.3.2.1 External actors

- **Client** - A registered client. The client also needs to have a valid Support Agreement with Xeratech. To be able to register an errand through the web interface, the client also needs an identifiable username and a password.

### 3.3.2.2 Internal actors

- **Telephone Operator** – A person within the support organization responsible for the answering and registering of incoming errands. The telephone operator cannot assign the errand to anyone; he or she simply acknowledges the Support Coordinator of incoming errands and fills in necessary metadata about the client.
- **Support Coordinator** – A support coordinator, or dispatcher, inspects incoming errands: whether they come from a telephone operator or directly from a client using the web interface. He or she then either rejects the errand, or delegates it to an Errand Consultant.
- **Errand Consultant** – The errand consultant is responsible for attending an incoming errand. Attending may include rejecting the errand (for reasons such as lack of Service Level Agreement for the specific errand, or that the errand consultant is unable to solve the issue of the errand). The errand consultant can also delegate the errand to another Errand Consultant, and hence also receive errands from other Errand Consultants.
- **Administrator:** Administrates all information in the System. The administrator has many roles: for instance, the administrator creates users and provides them with the appropriate permissions, construct and update the errand forms (the forms used for register an errand), makes reports and statistics etc.
- **Verifier** – Responsible for the verification of the ticket: when an *Errand Consultant* considers a ticket to be solved, the ticket first needs to be verified by the *Verifier*.

### 3.3.3 Entities

The Help Desk system consists of a handful of entities. An overview of these entities and the relation between them can be seen as an entity-relationship diagram in figure 8. The most important attributes of each entity will be explained later in this chapter.



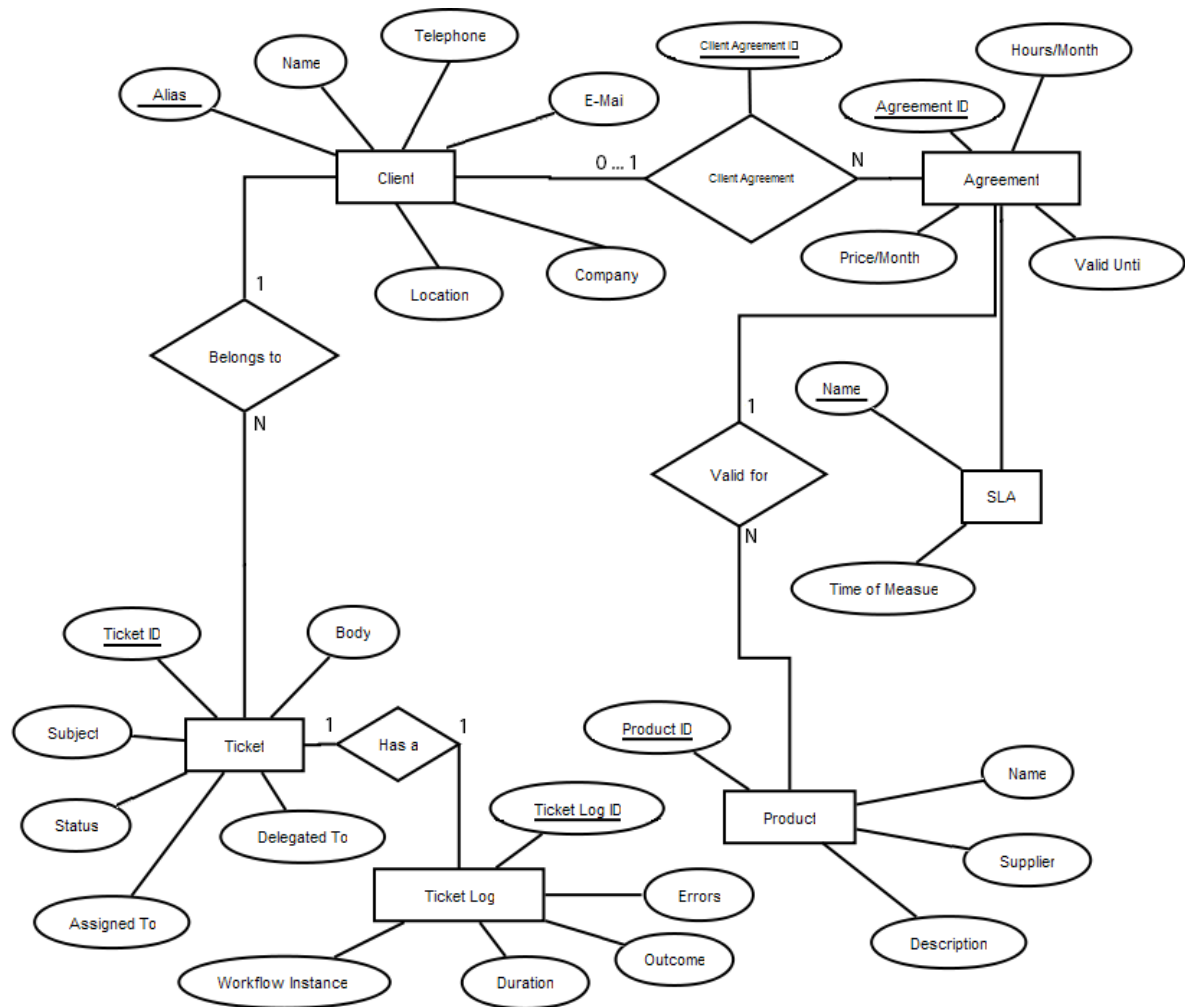


Figure 8: Overview of the entities in the Help Desk system.

### 3.3.3.1 Ticket

The ticket entity is the backbone of the system: this is where, for the majority of the help desk support team, the most work is situated. Each new errand, whether it comes from a client petition, a telephone operator or an errand consultant, is registered in a Ticket list. The ticket has a number of attributes.

- **Ticket Id** – Each ticket receives, when it is created, a unique identification number.
- **Subject** – The subject is a short introduction to the subject of the Ticket. This could be for example “The program xxx isn’t working” or “Need help installing xxx”.
- **Body** – The body contains a more thorough explanation of the nature of the errand. The body is also used by the errand consultants, which appends a response to the errand in the same body.

- **Status** – The status attribute is the single most important attribute for the help desk system, and is used by the Help Desk State Machine. The status attribute indicates which state the Help Desk State Machine currently has. The value of the attribute can be changed manually, for example, when an errand consultant wants to put an errand on hold, he changes the status attribute to *On Hold*. It can also be changed automatically by the system, for example from *New* to *In Progress*, when an errand consultant has received an errand and has confirmed that he or she has received it. The different statuses of the Help Desk State Machine are explained in detail in section 3.3.4.
- **Assigned To** – The Assigned To attribute contains the errand consultant which is currently assigned to the ticket.
- **Delegated To** – Contains the errand consultant that the ticket has been delegated to. When the errand consultant that is currently assigned to the ticket wants to *delegate* the ticket to another errand consultant, this attribute is used in order to do so. A delegation must be *verified* by the person who has received the ticket so that the ticket is not assigned to anyone who isn't able to start working with the ticket at the moment (more about verifications in section 3.3.5).

### 3.3.3.2 Client

The client is the only external actor in the system. Storing information about the client is necessary in order to be able to communicate with the client, but also for the client to be able to identify him- or herself before registering an errand online.

- **Alias** – This is a unique name for the client. Since a real name is not guaranteed to be unique, the client is provided with an alias (which however *can* be the same as the real name).
- **Client name** – The real name of the client.

### 3.3.3.3 Agreement<sup>1</sup>

The Agreement determines whether or not the client will receive support. Each client can have zero or more Agreements. If the client has zero Agreements, the client is automatically denied support. (This may very well occur, because the Agreements are usually expirable. In this case, the client will receive a notification saying that his or hers Agreement has expired, and will be asked to renew the Agreement.)

- **Agreement ID** – A unique identifier for the Agreement.
- **Agreement Paper** – An attachment of the written Agreement made by Xeratech and the client.
- **SLA** – Each Agreement have an SLA. The SLA state for instance the maximum time before a ticket needs to be started working with. (More about SLA in section 3.5.3)
- **Hours/Month** – Indicates the maximum hours per month a ticket will be worked with before Xeratech will charge extra.
- **Price/Month** – Indicates the price of the Agreement.
- **Valid Until** – The date for which the Agreement expires. (When the actual date is close to this date, the Client receives a notification about renewing the Agreement.)

### 3.3.3.4 Product

When an Agreement is established between a client and Xeratech, the Agreement states the products for which the Agreement is valid. In other words, in order for a client to receive support for a product, the client needs to have an Agreement for that specific product. The same product can exist in many agreements, and an agreement can contain many products.

---

<sup>1</sup> The Agreement is, in fact, divided into two separate Agreements: *Support Agreement* and *Maintenance Agreement*. The *Support Agreement* contains attributes such as how many hours per month the client can receive telephone support, whilst the *Maintenance Agreement* states for instance that the client will receive updates for a certain product during a certain amount of time. However, as for now, the *Maintenance Agreement* is neither defined nor prioritized. For that reason, the *Maintenance Agreement* will not be implemented, even though the system must be implemented in such way that the *Maintenance Agreement* can easily be added. So, henceforth, an *Agreement* refers to a *Support Agreement*.

- **Product ID** – A unique identifier for the Product.
- **Name** – The name of the Product.
- **Supplier** – The company behind the product. A product can be a product made by Xeratech, or it can be a product made by another company which Xeratech has agreed with to handle the support of. (The system will not differentiate these two types of Products.)
- **Description** – If additional information about the product needs to be specified, this is where it is done.

### 3.3.3.5 Ticket Log

An important function of the system is the ability to create a Ticket Log where useful information is stored. This in order for Xeratech to, out of this data, be able to create reports and statistics about the Help Desk system. The initial attributes will be explained below, but chances are that this list will grow larger, as there is difficult to foresee exactly which attributes that will be interesting when making reports and statistics. This is also something to have in consideration while implementing this function to the system.

- **Ticket Log ID** – A unique identifier for the Ticket Log.
- **Workflow Instance** – This attribute is a reference to the workflow instance associated to the Ticket. This attribute is not primarily for statistics and reports, but instead it makes it easier to find the corresponding Ticket, as it displays the Subject of the Ticket.
- **Total duration** – The amount of time it took for the workflow to go from *Initial State* to *Completed State*.
- **States traversed** – The number of states traversed before completion.
- **Errors** – The number of errors encountered.
- **Delegations** – The number of times the ticket was delegated to another errand consultant.
- **Outcome** – If the ticket, when completed, is *Solved* or *Rejected*.

The Ticket Log will also store information about how long time the ticket has been in each state.

### 3.3.3.5 SLA

The SLA determines the priority level of the ticket. The higher the priority, the shorter time the Help Desk support team will have to accomplish certain steps of the ticket workflow. Each Service Level Agreement has a number of attributes specifying these times.

- **Acute** – The highest priority.
- **Urgent** – The second highest priority.
- **Non Constraining** – The third highest priority. Non Constraining means that the problem for the product for which the errand refers to is not constraining the use of the product.
- **Beauty flaw** – This is the lowest priority. Beauty Flaw means that the product is still fully operative, but that the client has discovered some feature that should be updated, such as perhaps a misplaced button or a formulation error in a text.

The Service Level Agreements have names such as *High* or *Low*, where the High SLA has a higher priority, which is defined by lower attribute values than the *Low* SLA. When an errand is received, this priority is established. For instance, if a client has an *Acute* errand, the *Acute* attribute of the client's SLA is looked upon.

### 3.3.4 Help Desk State Machine Workflow

The workflow is, as mentioned in the introduction (section 3.1), the backbone of the Help Desk system. Each time a new ticket is created, it is associated with an instance of the *Help Desk State Machine Workflow*. This workflow handles all the steps necessary for the ticket to complete. The State Machine Workflow is (not all too surprisingly) comprised of different states. Each of these states handles its own logic. Figure 9 depicts an overview of the states, and the different transitions and events associated with them. This is a simplified view of the State Machine, and does not depict the different actions in each state. For example, each state has an *Initialization Action* and a *Finalization Action*, and the majority of the states have more activities. Below, each of the states of the workflow with belonging events is presented.

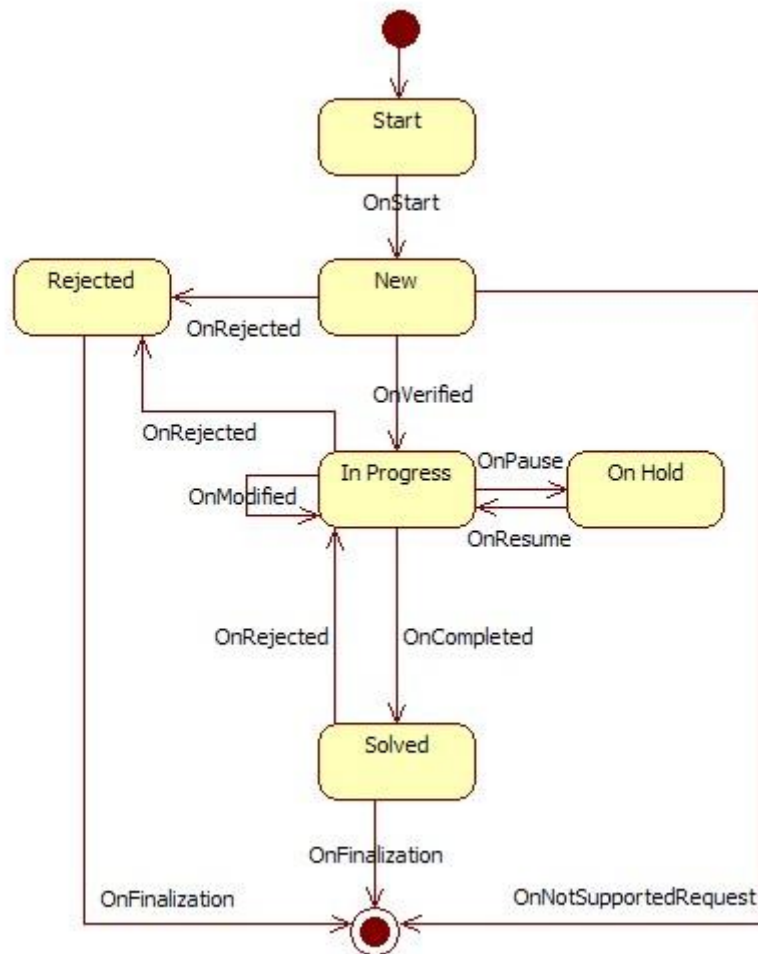


Figure 9: Overview of the states, transitions and events in the State Machine.

### 3.3.4.1 Start

The *Start* state is the first state in the workflow, regardless of input. The only purpose of the *Start* state is to read in the input from the ticket registration, and store the necessary variables in the workflow. The *Start* state will also initialize the *Ticket Log*.

- **OnStart** – The *OnStart* event is the only event in the *Start* State. It transitions the workflow to the *New* state. Also this is done regardless; each workflow instance will follow the exact same pattern.

### 3.3.4.2 New

The *New* state is, like the *Start* state, an automatic state in the sense that it doesn't depend on any input. Instead, every new workflow starts with *New* as status. The *New* state is responsible for sending out verification to the *Assigned To* errand consultant. The ticket

remains in the *New* state until the *Status* of the ticket manually changes or the *Assigned To* errand consultant verifies that he or she has received the ticket; the state is then changed to *In Progress*.

- **OnVerified** – Occurs when the *Assigned To* verifies the ticket. By verify it means that the errand consultant has decided to work with the ticket.
- **OnRejected** – Occurs if the *Assigned To* rejects the ticket.
- **OnNotSupportedRequest** – This occurs if the person who created the ticket, or the *Assigned To*, decides that the requested errand doesn't fall under the categories for which the client's Agreement. The ticket is then not processed. (Still it is necessary to register the ticket for statistic purposes.)

#### 3.3.4.3 In Progress

This is where the work with the ticket is made. The *In Progress* state is the state with most available events. The state itself doesn't do so much apart from indicating that the ticket is currently worked with. Nevertheless it is important that the ticket is in the *In Progress* state, to be able to see how much time the ticket has been worked on.

- **OnModified** – Happens when any of the attributes of the ticket is changed.
- **OnPause** – The *OnPause* event occurs when the *Assigned To* decides to pause the working with the ticket, the ticket then is transitioned to the *On Hold* state.
- **OnRejected** – The *In Progress* state also contains an *OnRejected* event, which works just as the *OnRejected* event in *New*.
- **OnComplete** – Occurs when the errand consultant considers the ticket to be completed. However, before completion, the ticket is transitioned into the *Solved* state to be verified.

#### 3.3.4.4 Solved

The solved state is an intermediate state where the ticket, in order to be *Completed*, first needs to be verified. When initiated, it sends out a *Solved Verification* (see 3.3.5).

- **OnFinalization** – Occurs if the ticket has been *verified*. The ticket is then considered *Completed*.
- **OnRejected** - Occurs if the ticket has been *rejected*. In this case, the ticket is transitioned back to the *In Progress* state. (Hopefully with a comment as to why the *Solved Verification* has been rejected.)

#### 3.3.4.5 Rejected

The *Rejected* state is a finalization state, and once the ticket reaches this state, the next state will always be *Complete*.

- **OnFinalization** – The only event in the *Reject* state. It merely forwards the ticket to *Complete*, and indicates in the *Ticket Log* that the ticket was completed as *Rejected*.

### 3.3.5 Notifications (sub-workflows)

During the lifetime of a ticket, several notifications, affirmations and verifications needs to be made. Common for all verifications is that they are sent to another person within the Help Desk Support Team (see Internal Actors in section 3.3.2.2), and that the person receiving the verification responds with some form of *Reject* or *Approve*. These can be seen as *sub-workflows* to the main workflow. Every notification shares all but one field: the *Verification* field. This field is customized to prevent ambiguities.

The different notifications are listed below.

- **Assigned To Affirmation** – Occurs when a new ticket is registered. The verification is sent by the person that created the ticket, and the receiver is the person that has been assigned to the ticket. The main workflow is halted until an *Assigned To* accepts the assignment.



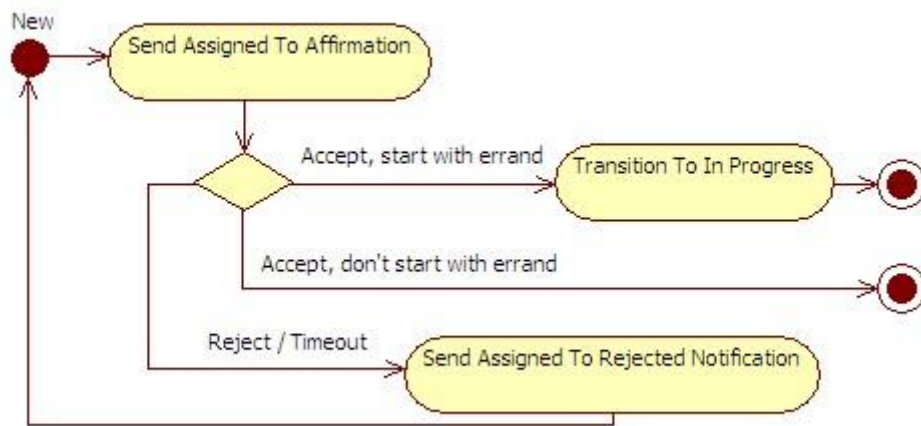


Figure 10: *Assigned To Affirmation -and Rejected Notification activity diagram.*

- **Assigned To Rejected Notification** – This can be seen as a sub-workflow to the *Assigned To Verification*, and occurs when the person assigned to the ticket rejects it. Then a new notification needs to be sent out to the creator of the ticket saying that he or she needs to assign the ticket to another person. This also occurs when the notification *times out*.
- **Delegated To Affirmation** – Occurs each time a ticket is delegated to a person. The current *Assigned To* is the sender, and *Delegated To* is the receiver. The main workflow is not halted until an affirmation is received. Instead, the current *Assigned To* works with the ticket until the affirmation, or rejection, is received. In case of a rejection, the *Assigned To* continue working with the ticket, or *delegates* the ticket to another Errand Consultant.

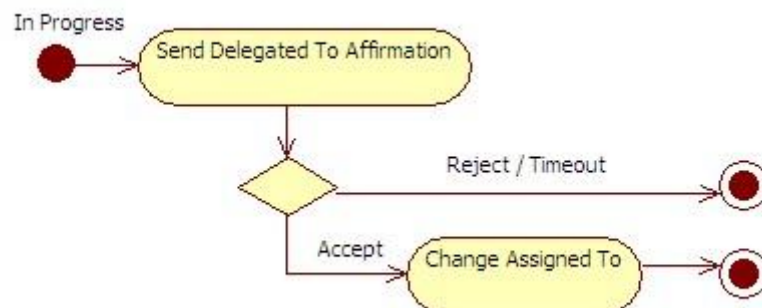


Figure 11: *Delegated To Affirmation activity diagram.*

- **Solved Verification** - This verification is sent when an errand consultant considers that the ticket has been completed. The ticket then needs to be *verified*. This is done by the *Verifier* (see section 3.3.2.2: Internal actors).

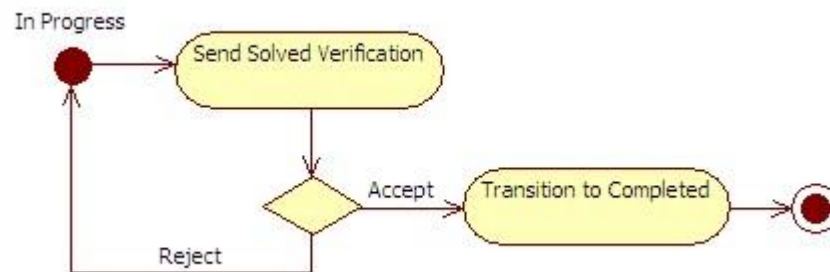


Figure 12: Solved Verification activity diagram.

All these verifications will, if the receiver does not answer the notification in time, eventually *timeout*. Hence, each sub-workflow will have a *timer* associated with it (see section 3.3.5).

### 3.3.6 Timers

Timers are an important part of the system. There are three types of timers: timers associated with the *Agreement*, timers associated with the *Ticket Log* and *Workflow Update Timer*.

- **Agreement Timers** – These timers are started and stopped depending on the actual *Agreement* indicated on ticket registration. The response time of, for instance, the *Assigned To Verification* depends on the *priority* of the ticket. For example: a ticket has priority *Urgent*, and because of that, the *Assigned To* has (for example) six hours before he or she needs to have verified and started working with the ticket. If the priority would have been *Non Constraining*, the time would have been perhaps ten hours instead of six (depending on the *Agreement*).
- **Ticket Log Timers** – The purpose of these timers is to log the time the ticket stays in each state of the state machine, so that statistics can be made. This is of special interest in the *In Progress* state, as this measure the time the ticket has been worked with.
- **Workflow Update Timer** - This is a timer associated with the entire *Ticket List*, and each of its elements. There will exist one and only one *Workflow Update Timer*, and

this timer will be external, i.e. not implemented in the workflow. The only thing this timer does is to update each item in the *Ticket List*, so that each item can see if any of the *Agreement Timers* have timed out, so that *Notifications* can be sent out.

### **3.4 Chapter summary**

The purpose of the Help Desk Support System is to be able to process an errand from an external actor (i.e. a client), assign it to a person within the Support Team, and to facilitate the rest of the work with the errand.

The backbone of the Help Desk Support System is the *State Machine Workflow*, which handles all the internal parts of the Help Desk; it stores the attributes of the client, it is responsible for alerting the person within the support team currently assigned to the ticket, as well as alerting the newly assigned person in case of a *delegation*, and it keeps track of how long time the ticket has stayed in each state. It also gathers useful information, so that statistics and reports can be made out of this information.

## 4 Implementing the help desk system

---

*The implementation of the Help Desk System is based on the design made in the previous chapter, and the implementation of this design is going to be explained in this chapter. The different programs needed to manage the system will also be presented.*

---

### 4.1 Introduction

The implementation of the Help Desk system is divided into three parts: the *Ticket Registering* part, the *State Machine* part, and the *Administration* part. These three are the basic components of the Help Desk system. Thus, this chapter will have the same structure, starting with the *Ticket Registering* in section 4.4, the *State Machine* part in section 4.5, and the *Administration* part in section 4.6. Before describing these however, section 4.3 *Implementing in SharePoint* will deal with all the built in features from SharePoint used in the implementation.

### 4.2 Setup

There are several programs that need to be installed and properly configured in order for the SharePoint 2007 Workflow to work properly. A short description of these programs will be presented (if not already introduced in previous chapters). If needed, the description will also contain a short motivation as to why that particular program has been selected. However, these are the programs used in this particular application, and are surely to vary in other, similar applications.

#### **4.2.1 Microsoft Virtual PC**

A virtual machine containing the SharePoint Site will be setup. A virtual machine is advantageous in many ways. For instance, it allows developers to have a separate environment for every project. This means that every project can be customized according to necessity. This is not the only advantage; some other advantages are [10]:

- **Sandboxing**- The ability to quickly set up a test environment, independent of the standard environment.
- **Testing**- Simulation of multiple operating systems and software version is possible within a virtual machine.
- **Rollback**- Microsoft Virtual PC makes it easy to restore files.

#### **4.2.2 Microsoft Visual Studio 2005**

The Microsoft Visual Studio 2005 is where the most of the work of the system is going to take place, and it's where the coding of the workflow is going to be made.

#### **4.2.3 Workflow extensions for Microsoft Visual Studio 2005**

This workflow extension enables Microsoft Visual Studio 2005 to use the SharePoint Sequential and the SharePoint State Machine workflow templates. The State Machine workflow template is going to be used implementing the help desk system.

#### **4.2.4 Windows SharePoint Services 3.0**

Only WSS will be used in the implementation (i.e. MOSS will not be used), see chapter 2.3.3 for more information about WSS and MOSS.

#### **4.2.5 Microsoft Office SharePoint Designer 2007**

The SPD is very useful for making cosmetic changes to SharePoint Web sites, such as adding a button, change the CSS layout or adding an existing web part.

#### **4.2.6 Windows Server 2003**

The Windows Server 2003 is used as the server for the system. It resides in the virtual machine. In order for it to work with SharePoint, the IIS and ASP .NET 2.0 needs to be installed. (To enable ASP .NET 2.0 in IIS, use the `aspnet_regiis -i` in the command prompt.)

#### **4.2.7 Hardware**

When using a virtual machine, the memory aspect is important. Since the host machine and the virtual machine will need to share the available memory, memory is critical. From experience, a total memory of less than 2 GB usually will not suffice. For this project, to be able to run applications like Visual Studio, a laptop with 4 GB of memory is used, allocating 2 GB to the guest operating system (on the virtual machine).

### **4.3 Implementing in SharePoint**

Before describing the different parts of the implementation, it is worth mentioning how some of the things in SharePoint are implemented, as this applies for all the different implementation parts. Implementing these in SharePoint is not so much a question of programming. In fact, these things come out-of-the-box in SharePoint, and can be made without writing a single line of code.

#### **4.3.1 Site**

The whole system (except for the client external web site) resides within a SharePoint site collection. This site collection is created on a virtual machine. Neither the root web (the highest directory in the SharePoint Site) nor the sub-sites in the site collection, except the *Help Desk Site*, are manually created. Instead, a template has been used that provides a basis of several sites and lists. This has no importance for the final system; the template only serves as a platform (or playground) for the development of the system and will, once the system is

finished, be discarded and replaced with the actual Xerotech Internal Web (in other words, the Help Desk Site will be, once it's finished, integrated with Xerotech Internal Web).

Creating a site in SharePoint is really simple: with two clicks, a new site is created and added to the collection of sites. The *Help Desk Site* is created in this way. The design of this page has not been changed; it automatically inherits the appearance of the parent web, and that has been sufficient (as the design of the Xerotech Internal Web will most likely differ from the design of the template used, there has been no meaning of changing the design).

A site can be created using a template. The *Help Desk Site* is created using the *Team Site template* (see Appendix A). Figure 13 depicts the *Help Desk Site* and more concretely the *Help Desk List*. (The other lists can be seen in the left column. The majority of these will however only be available and seen by the *administrator*.)

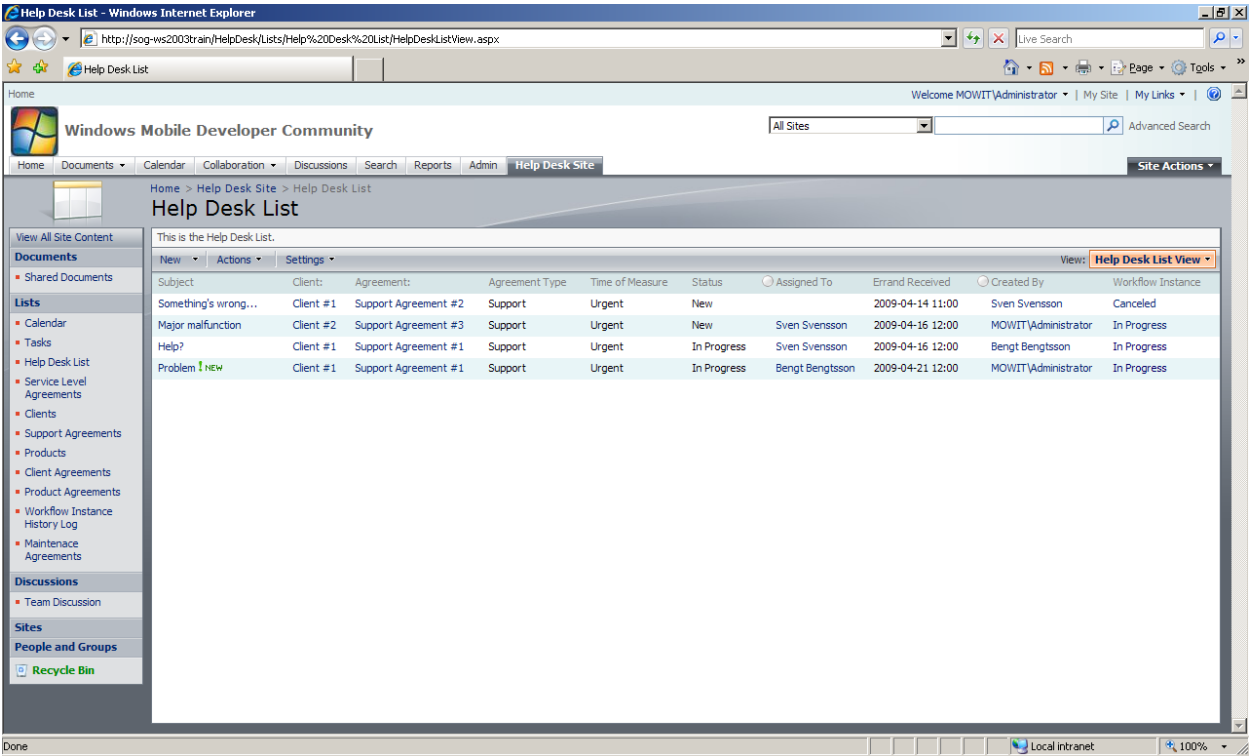


Figure 13: The Help Desk Site.

With the *Team Site template*, several lists are created automatically. This includes the *Announcements*, *Calendar*, *Links*, *Tasks*, and *Discussion* lists. The list template defines the columns that the list will contain and the views of the list items that are displayed (more about

Custom Lists in section 4.3.2). However, the only list that will be used from the automatically created lists is the *Task* list. See section 4.3.3 for information.

### 4.3.2 Custom Lists

Custom Lists in SharePoint are easy to create. Just as with a *site*, a *custom list* can be created with one or two mouse clicks. Along with the *Task* list, all the lists used for the *Help Desk* system are created as custom lists. These lists are the *entities* of the system (see chapter 3.3.3). The lists, and the relationship between them, are depicted in figure 8.

Each list, whether it's a custom list or a list defined by a list template, consists of three parts: *items*, *columns* and *Views* [3]. See figure 14.

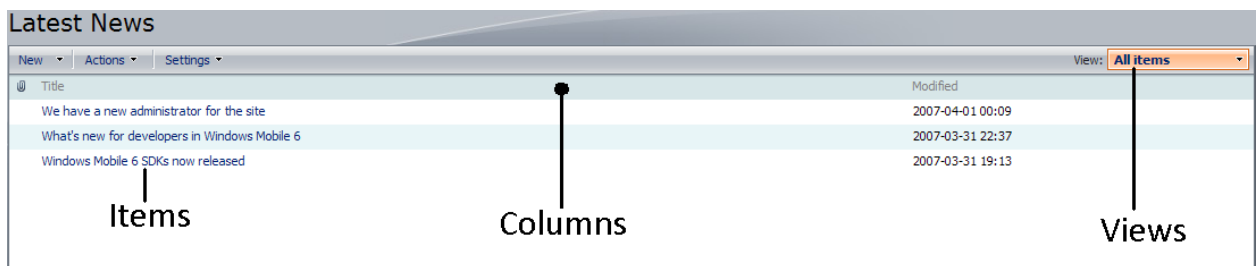


Figure 14: The parts of a SharePoint List.

- **Item** - Items are the rows of data in a list. Users add new items or change existing ones. The adding of an item is done by filling in a *form*. See section 4.3.3 for more information about *forms*.
- **Column** - Columns define the types of data that a list contains. Columns are also called *fields* in the Microsoft documentation. All custom lists contains by default the columns *Created*, *Created By*, *Modified* and *Modified By*, where *Created* and *Modified* contains the date and time when the list item was created and modified, respectively.
- **View** - control what columns are displayed, how they appear, and what filters or grouping are applied to the rows. This is *very* useful when it comes to displaying different information to different users, as well as grouping information in different



ways to facilitate finding items based on a certain criteria. An example of a view might be *My Errands*, *All Errands With Status In Progress* etc. The *view* will be described more out of this perspective in section 4.3.5.

### 4.3.3 Form

Creating a new item in a list is done by filling in a form. For every list that is created, three form pages and one view page are created [2]:

- **DispForm.aspx** – When a User clicks on an item and chooses *Display item*, the *DispForm.aspx* page will be shown.
- **EditForm.aspx** - When a User clicks on an item and chooses *Edit item*, the *EditForm.aspx* page will be shown.
- **NewForm.aspx** - When a User creates a new item, the *NewForm.aspx* page will be shown.
- **AllItems.aspx** - A fourth file is also created by default, this is the default data view that displays the items in the list.

The fields in a form depend on the columns in the list: these columns will automatically be presented in the list. A list can also use *custom item forms*. Custom item forms are done by modifying one or more of the four pages described in the above list, or substituting them with new pages and then place these new pages in the same folder as the list. Custom item forms will, however, not be used in any of the lists created in the *Help Desk Site*.

An out-of-the-box form can be customized to some extent: it is possible to specify which fields that are hidden, which fields that are optional and which fields that are required. The major disadvantage of using out-of-the-box forms is that each of the three form pages will contain the same fields. This is not always desirable. This is especially important in the *Help Desk List* (or the *Ticket* list), because this is the only list where anybody else than the *Administrator* will be able to create and edit the list items. For instance, in the *Help Desk List*, the *New Item* form should not contain the same fields as the *Edit Item*, as some of the fields, such as the *Subject*, should only exist in the *New Item* because its value never should change,

once it's provided in the *New Item* form. This can be achieved using a *feature* called `SPListDisplaySetting`<sup>2</sup>.

Table 1 shows which of the fields in the *Help Desk List* that exist in which form.

*Table 1: The forms in the Help Desk List and the fields that they contain.*

	<b>New</b>	<b>View</b>	<b>Edit</b>
<b>Agreement</b>	X	X	
<b>Agreement Type</b>		X	
<b>Assigned To</b>	X	X	
<b>Body</b>	X	X	X
<b>Client</b>	X	X	
<b>Delegated To</b>		X	X
<b>Errand Received</b>	X	X	
<b>Last Update</b>		X	
<b>Status</b>	X	X	X
<b>Subject</b>	X	X	
<b>Time of Measure</b>	X	X	
<b>Verifier</b>	X	X	

#### 4.3.4 Tasks

Tasks are an important part of the workflow: it can be seen as sub-workflows to the main workflow. The tasks are linked to the workflow instance that created them, and can therefore notify their “parent” workflow when they change or get deleted so that the main workflow can take appropriate action.

Whenever an important step has been made, or is to be made, that requires verification by another person, tasks are used. These tasks are created programmatically inside the main

<sup>2</sup> `SPListDisplaySetting` is a setting that provides advanced settings to customize list form rendering in new, display and edit mode [25].

workflow. Figure 15 depicts one of the three types of tasks: the *Assigned To Verification Edit* form (the rest of the forms can be found in Appendix A).

Attach File | Delete Item | Spelling... \* indicates a required field

**Comments**

I don't have time today...]

Add additional comments that will be sent with the verification.

**Assignment Status \***

Accept, start with errand.

Accept, don't start with errand.

Reject.

Select the response to the assignment of the errand.

Version: 2.0  
Created at 2009-04-07 16:02 by MOWIT\Administrator  
Last modified at 2009-04-07 16:03 by MOWIT\Administrator

OK Cancel

Figure 15: The *Assigned To Verification Edit* form.

As mentioned in section 4.3.3, no custom item form is used. So how can the three types of workflow have different fields? The answer to this is to use *content types* [26]. A content type is a reusable collection of settings that can be applied to a certain category of content. Since the fields, the *metadata*, of a content type can be changed, the fields of the forms automatically changes along with the fields of the item. One list can have several content types. Hence, the *Task* list has four content types, one for each of the four types of notifications.

The lifetime of a sub-workflow reaches from the instance they are created until the receiver of the notification answers and submits the answer, *or* when the notification *timeout*. (See chapter 3.3.5 for a description of the actions taken by each type of *Notification* when a *timeout* occurs.)

#### 4.3.5 Permissions, Groups and Roles

Each person within the Help Desk team is a member of one or more groups. Each group has a *Permission Level* that contains a number of *Permissions*. This is how the role of the different

actors is decided: based on what group they belong (and consequently, what permissions they have). This is very useful since it then allows all the members of the Help Desk team to work in the same interface. SharePoint will automatically disable the actions not permitted for each user, either by simply removing the button for the not permitted action, or displaying a “Not permitted” text whenever the user tries to access a section or a list for which he or she doesn’t have permission. This is the same for the *view* in the *Help Desk List*.

There are three different types of permissions: *List permissions*, *Site Permissions* and *Personal Permissions*, adding up to a total of thirty three permissions. (A list of all the *Permissions* plus a short description of them can be found in Appendix A.)

SharePoint comes with four built-in permission levels. These will not be used in the system. The Help Desk system consists of five groups. List 2 shows each of the permission levels and a brief description of it. Each permission level has the same name as the group it is associated with.

*Table 2: The permission levels in the Help Desk system.*

<b>Permission Level</b>	<b>Description</b>
Administrator	Has full control (all permissions). Including create subsites, create groups, apply themes and add clients.
Errand Consultant	Can add, edit, view and delete errands. Can open and view pages in the web site. Can also edit and manage personal user information.
Errand Coordinator	Can add, edit, view and delete errands.
Telephone Operator	Can add and view errands. Cannot assign the errand to anyone (this has to be done by the Errand Coordinator.)
Verifier	Can receive Solved Verifications. Can edit and view errands.

Note that these permission levels can be combined. For instance, a person can be a member of both the Errand Consultant group *and* the Verifier group.

### 4.3.6 Fields

Most of the fields in SharePoint are not unique for SharePoint, and do not need further presentation. However, some fields need an introduction. List 3 shows the field types of the columns in *Help Desk List*.

Table 3: *The fields in the Help Desk List and their types.*

<b>Field</b>	<b>Type</b>
<b>Agreement</b>	Connected Lookup Field
<b>Agreement Type</b>	Choice
<b>Assigned To</b>	Person or Group
<b>Body</b>	Multiple lines of text
<b>Client</b>	Connected Lookup Field
<b>Delegated To</b>	Person or Group
<b>Errand Received</b>	Date and Time
<b>Last Update</b>	Date and Time
<b>Status</b>	Choice
<b>Subject</b>	Single line of text
<b>Time of Measure</b>	Choice
<b>Verifier</b>	Person or Group

#### 4.3.6.1 The Person or Group field

The approach, to distinguish the different roles of the persons by adding them to different groups, is further supported by the so called *Person or Group* field in SharePoint. When a *Person or Group* column is created in SharePoint, the group to *choose from* is selected. For instance, in the *Ticket List*, the *Assigned To* field is a *Person or Group* field, where only the *Errand Consultant Group* members are available for choosing.

#### 4.3.6.2 The Connected Lookup field

The *Connected Lookup Field*<sup>3</sup> is not a built-in type in SharePoint, but an *external feature*. SharePoint come with a built-in *Lookup* field. This field has a shortcoming though; it does not offer *connected lookups*. By *connected lookup* means that a field (in our case, a drop-down list), can be populated depending on another field. The two columns in the *Help Desk List* with this type: the *Client* and the *Agreement*, are connected to each other. The result of this is that when a new errand is registered, and the client is selected, the *Agreement* field will update, and the drop-down list will only contain the agreements for that particular client. The list that the fields are connected to is the *Client Agreement* list which is the relationship between the *Client* list and the *Agreement* list.

## 4.4 Ticket Registering

The registering of a ticket can be made in two ways: internal or external. An internal registration can be made by anyone in the *Help Desk* team. However, depending on who made the registration, the ticket will be processed a little different. The second way, the external, is made by a client from an external web page.

### 4.4.1 Internal Registration

Eventually, all tickets registered by anyone within the *Help Desk* team will be processed in the same manner. However, at first, the registration differs a little. Below is a list of the groups, describing how the registration is processed when a member of this group register a ticket.

- **Telephone Operator Group** – Any ticket registered by a member of this group will not have the *Assigned To* and *Verifier* field filled in. This is because the *permission level* of this group does not allow its members to do that. Instead, whenever a ticket is register by a member of this group, the *Support Coordinator Group* is alerted that a new ticket has been received. The *Support Coordinator* can then assign the ticket to whom they see fit. (This is not considered a sub-workflow. The reason to this is that the alert only consists of an e-mail, and not a task to reply to.)

---

<sup>3</sup> Connected Lookup Field offers connected lookup between two fields in a SharePoint List [27].

- **Support Coordinator Group** – When a ticket is registered by a member of this group, no alert to any other group needs to be sent (except of course a notification to the *Assigned To*, if that is specified). The *Verifier* field will however not be accessible. (This is because the *Verifier* is not established for each ticket.)
- **Errand Consultant Group** – The *Errand Consultant Group* can also register tickets. The members of this group have the same *permissions* when it comes to creating, editing and viewing tickets as the *Support Coordinator Group*.

## 4.4.2 External Registration

### 4.4.2.1 Web Site Registration

Errand registration can be made by the clients themselves. This is made from the *Client Support Web*. This is a simple interface for the clients, where the only thing that is required is for the clients, in order to register a new errand, to login with their alias and password (specified in the *Clients List*), and then to specify the *Subject* and the *Body* of the errand. Figure 16 shows the page for registering a new errand.

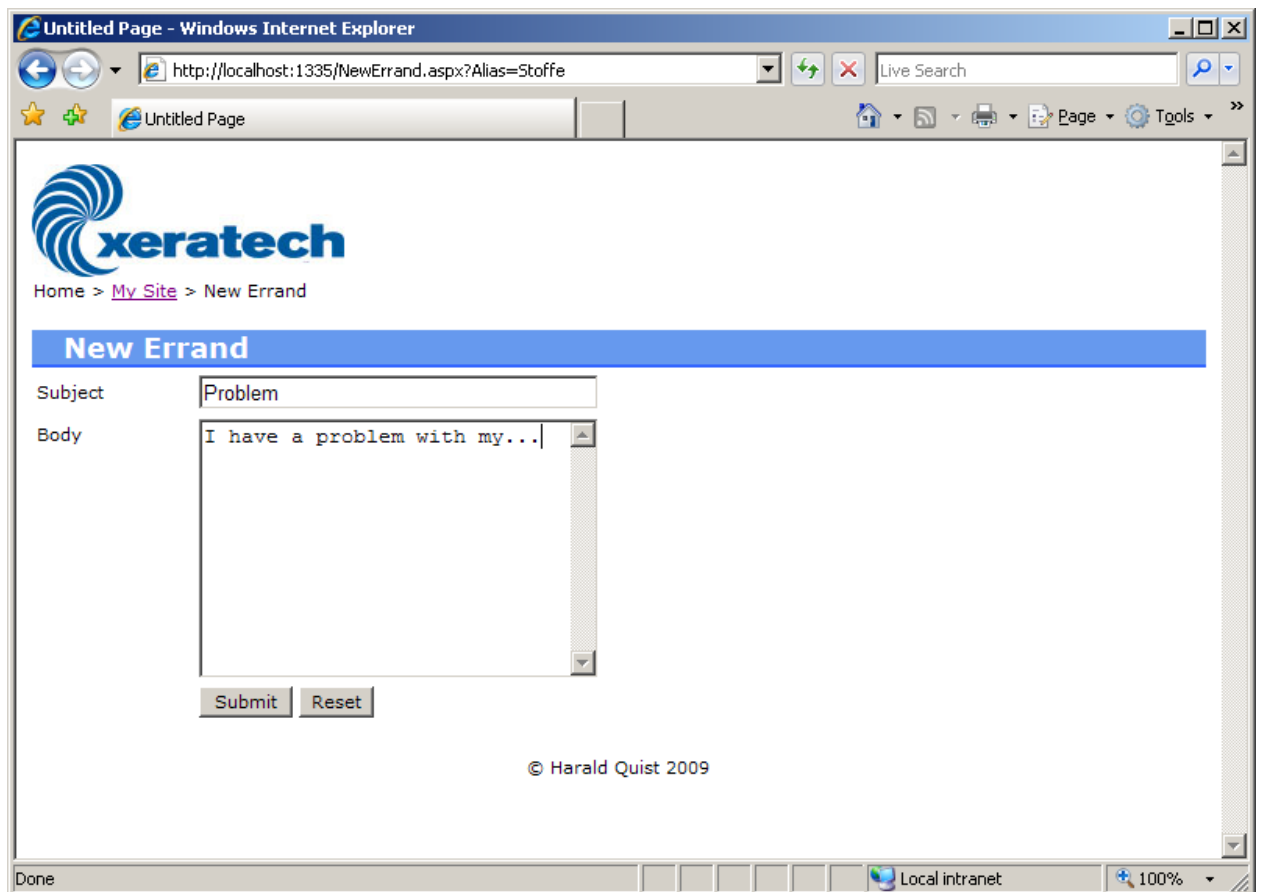


Figure 16: Register a new errand in the Client Support Site.

When the client *Submits* the errand, the errand will be automatically added to the *Help Desk List*. These errands will then be handled in the same way as errands registered with no *Assigned To*, since the client does not have the ability to assign an errand to anyone.

In the same site, the clients also have the ability to view the status of his or her other tickets, if any.

#### 4.4.2.2 Telephone Registration

When a client calls a member of the support team, the ticket is registered in real time. Since the client (if it indeed is an existing client that calls) and the information about the client already exist in the Help Desk database, registration by phone will not be a difficult task. The support team roles are applied in the same manner for telephone registrations. (For example, the Telephone Operator group will still not be able to specify the *Assigned To*.)



## 4.5 The State Machine

The state machine workflow is implemented in Visual Studio 2005. In SharePoint terms, the workflow is implemented as a *feature*. This *feature* is then associated to a certain list, namely the *Help Desk List*. Figure 17 shows the states of the workflow implemented in Visual Studio 2005.

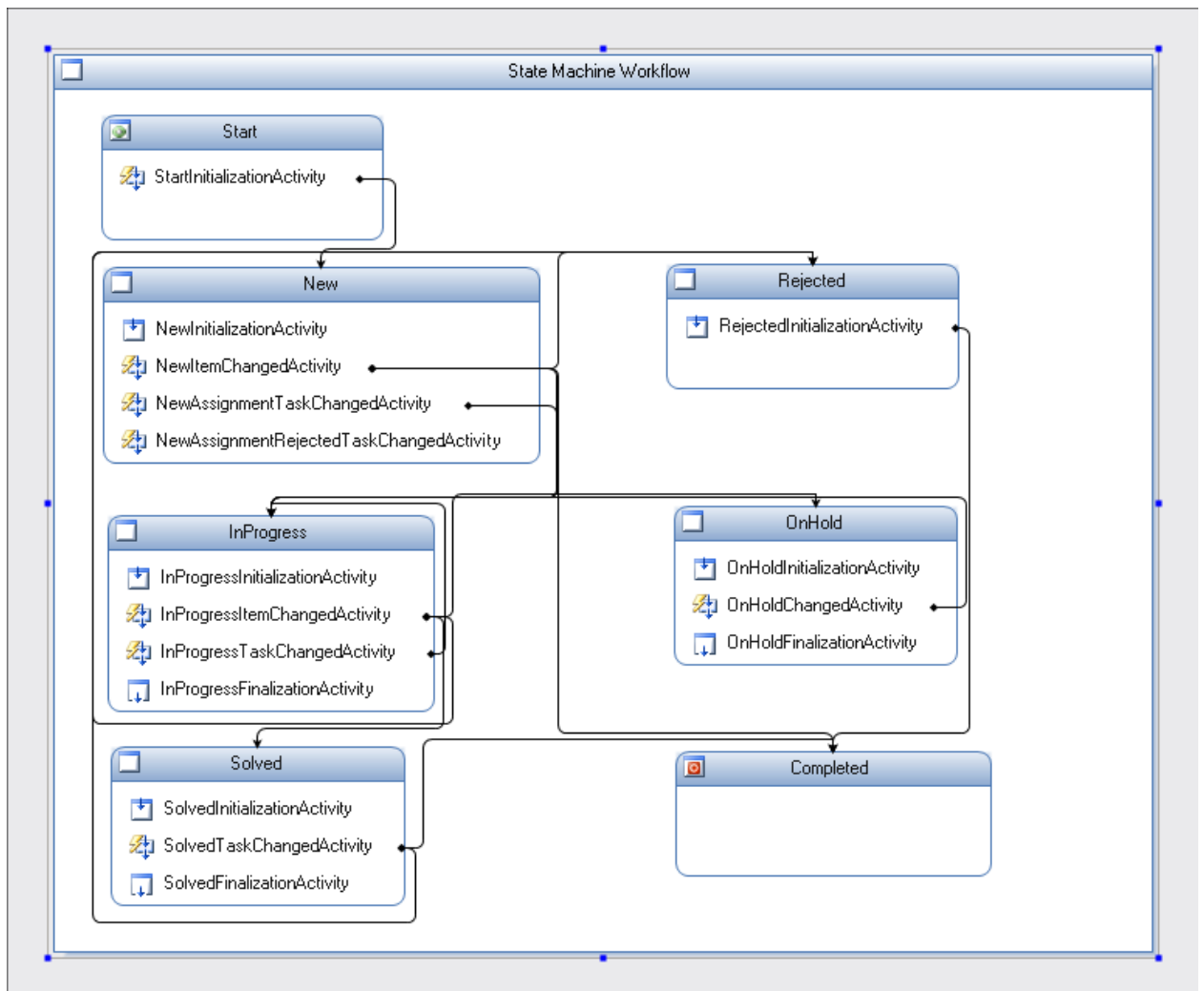


Figure 17: Diagram of the states and activities in the State Machine.

These are the same states, with the same transitions, as shown in figure 9. Figure 17 however, shows how the state machine is constructed programmatically. The activities within the states are also depicted.

## 4.5.1 State Machine Activities

There are a total of three different activities used in the state machine. Each of these is named in a consistent way.

### 4.5.1.1 <State Name>InitializationActivity

This activity occurs when a state initializes. The things that needs to be done on initialization is placed in this activity. For example, every state uses this activity to write to the *Log* that the state has initialized. Also, each state uses this activity to change the *Status* field in the *Help Desk List* to the current state. It is also used to start the timer for the current state.

### 4.5.1.2 <State Name> EventDrivenActivity

This activity is the most used activity: within this activity, all the *events* are placed.

### 4.5.1.3 <State Name> FinalizationActivity

This activity is called whenever a state finalizes, i.e. it is the last activity that is called in a state before the state transitions. It is used mainly to stop the timer of the current state.

## 4.5.2 State Machine Events

Two types of events exist in the state machine: the *OnWorkflowItemChanged* event and the *OnTaskChanged* event. These are implemented within an *EventDrivenActivity*.

### 4.5.2.1 *OnWorkflowItemChanged*

This event occurs when the associated item in the *Help Desk List* is changed in any way. Every state that uses this event calls the function *CheckFieldChanges()*. This function checks all the fields in the *Help Desk List* against the values stored in the *Ticket* class (see section 4.5.3). Figure 18 shows this function in pseudo code.

```

String CheckFieldChanges () {
    String updateFields = ""
    For each(Field f in helpDeskList.Rows[wfInstance.Index() ].Fields) {
        If (ticket.Fields[f.Index() ].NotEquals (f.Value())) {
            updateFields += f.Value()
            // Do appropriate updates.
        }
    }
    // The string with the updated fields will be logged.
    Return updateFields
}

```

Figure 18: Pseudo-code for the *CheckFieldChanges()* function.

In the *CheckFieldChanges* method, showed in figure 18, each field of the workflow instance (named *wfInstance* in the figure) is checked against the saved values in the ticket. A change is made if any value differs from the saved value in the ticket.

Every event of the type *OnWorkflowItemChanged* is implemented within an activity with the name **<State Name>ItemChangedActivity**.

This activity is the most frequently used activity in the workflow: each time an item in the Help Desk List changes, its associated workflow will trigger this activity no matter what have been changed. Therefore, it is important to check each field in the item available for the user (the Help Desk List has more fields than those seen by users, but they are not relevant since they cannot be changed if not explicitly changed programmatically).

As an example of what these activities look like in Visual Studio 2005, consider figure 19. Figure 19 shows the *ItemChangedActivity* activity in the *In Progress* state. Since the *In Progress* state is the state where the actual work with the errand takes place, the event within this activity is going to be triggered several times within the lifetime of an errand. What it does, in short, is that it first checks if anything has been changed (the yellow rectangular box in figure 19 shows the event), then the workflow branches with an if/else statement (see the green lines). The first branch (the left one) checks to see if the *Delegated To* field has been changed. If it has changed, it proceeds and sends a *Delegated To Notification* to the person stated in said field. (Note that no check to see if the person in this field is valid needs to be done, since the only persons available to choose are persons from the *Errand Consultant Group*, which all are valid. See section 4.3.5: Permission, Groups and Roles.)

The second branch checks to see if the *Status* field has been changed. If so, the workflow needs to change state. This is done with a nested if/else statement (the *IfInProgressStateChanged*). Note that not all states are checked for: only the valid states (depicted in figure 9) are available. This is an easy way to make sure no forbidden transition takes place.

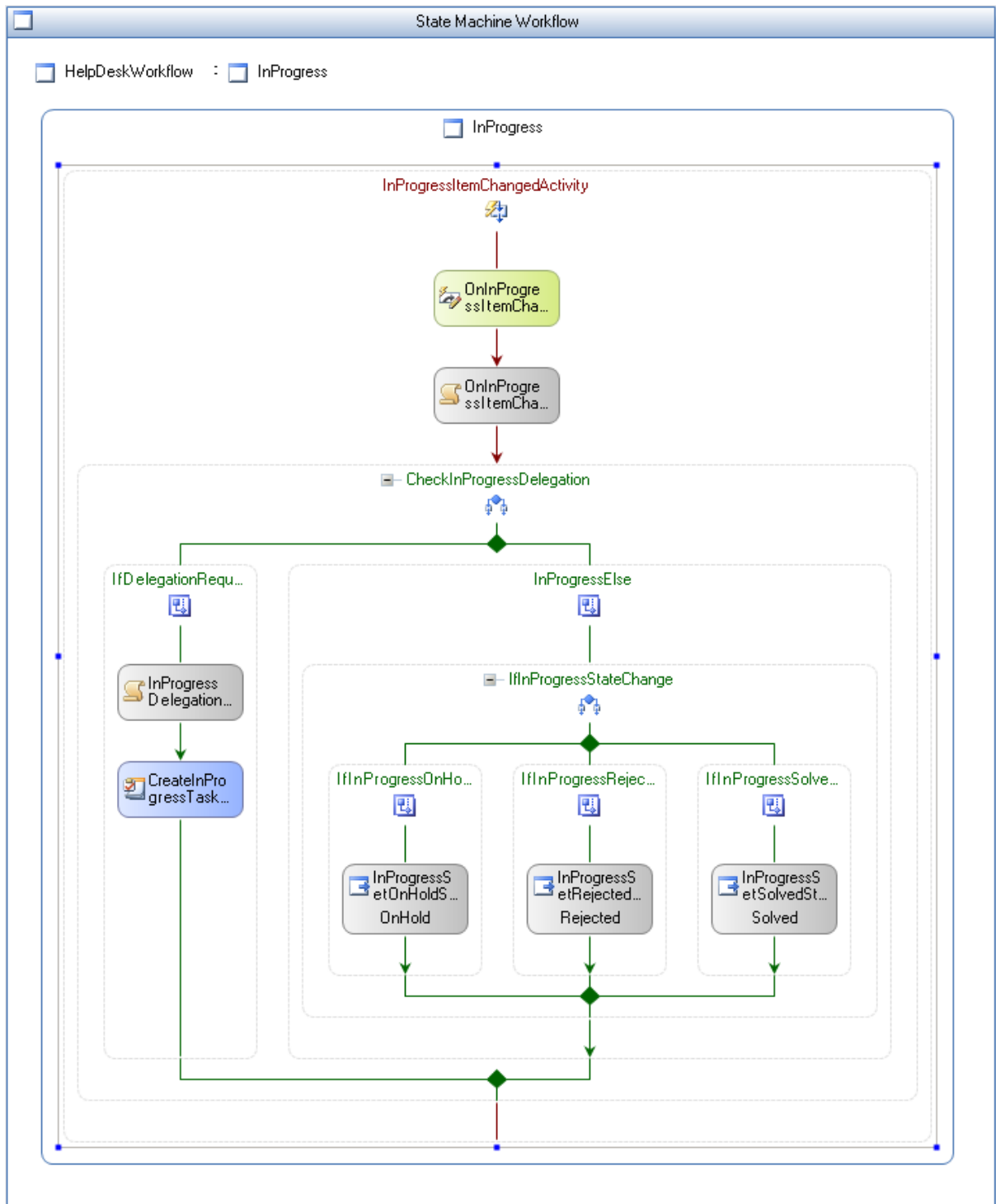


Figure 19: The InProgressChangedActivity.

#### 4.5.2.2 OnTaskChanged

This is an event that occurs when a task (sub-workflow) changes within the state. This event is, of course, only available if a *CreateTask* has been made at some earlier stage within the state. The activity behaves in similar way as the *<State Name>Changed* activity. The

difference is that this activity checks the changes of the associated task instead of the whole workflow. A change in the task equals a response from the person receiving the task. For example, the *TaskChangedActivity* for the *Delegated To Affirmation* task checks to see which answer has been received. Every event of this type is implemented within an activity with the name *<State Name><Task Name>TaskChangedActivity*.

### 4.5.3 State Machine Classes

When an errand is registered, it is automatically associated with an instance of the *Help Desk State Machine*. For each ticket that is registered, a number of classes is started to facilitate the processing of the ticket. Figure 20 depicts the classes and the relationship between them.

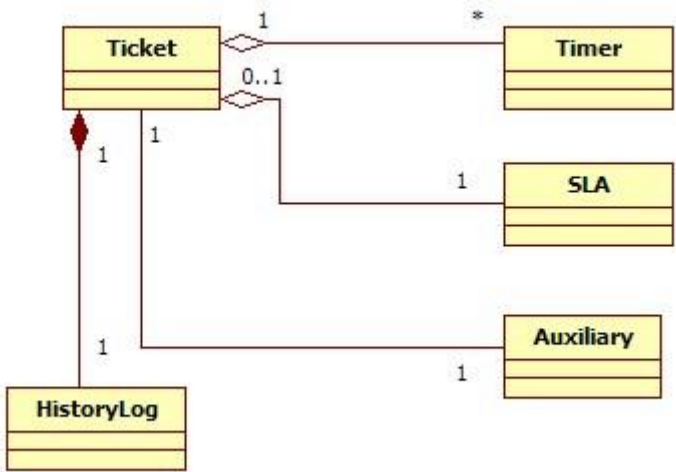


Figure 20: Diagram of the classes in the State Machine.

- Ticket** – The *Ticket* class stores all the variables needed from the *Help Desk List* and the other lists needed (The *Client Agreement List*, for instance). This makes it easier not only to overview the system, but to make checks of the sort “if the stored value is not the same as the value in the list, handle accordingly based upon these changes”.
- Timer** – The *Timer* class provides the *Ticket* class with timers. These timers are not timers in a traditional way (i.e. that an event handler increments a counter every *X* seconds). Instead, each new instance of the class stores the time when the timer is going to time out, and also has the ability to stop the timer and start it again. The reason why they aren’t implemented as regular timers is because SharePoint won’t

allow *non-serializable* object to be stored in the workflow (see section 5.1.1.3 for an explanation of *serialization*.)

- **SLA** – The variables from the *Service Level Agreement List* are stored in this class.
- **HistoryLog** – The *History Log* is responsible for storing each change of state, the time the workflow has been in each state etc.
- **Auxiliary** – The main task of the *Auxiliary* class is to check the answers from the different sub-workflows in the system. The answers all have different text, so this class translates the answers to an *enum* value (an integer).

#### 4.5.4 Timer Jobs

One thing a State Machine Workflow cannot provide is the ability to update itself (its instances) on a regular basis. Without a *Timer Job*, a workflow is only updated when an *event* is triggered. For *notifications* this is not enough, since the workflow instance for a certain ticket perhaps isn't updated in a long time, and a *notification* needs to be sent within twenty hours. Because of this, a *Timer Job*<sup>4</sup> is implemented. The *Timer Job*, named *WorkflowItemsUpdateTimerJob*, does just one thing: it updates all the items in the *Help Desk List*. It is implemented as a *feature* in SharePoint. Figure 21 shows the pseudo-code for the *Execute* function of the *Timer Job*.

---

<sup>4</sup> Timer Job – Handled in WSS under Central Administration. Timer jobs are defined by using a single class that inherits from the `Microsoft.SharePoint.Administration.SPJobDefinition` class.

```

Void Execute() {
    SPList helpDeskList
    For each(SPWeb web in content.Sites.AllWebs) {
        If(web.Name.Equals("HelpDesk")) {
            For each(SPList list in web.Lists) {
                If(list.Title.Equals("Help Desk List")) {
                    // Help Desk List found.
                    helpDeskList= list
                }
            }
        }
    }
    If(helpDeskList.NotEquals(null)) {
        // Update all items in list
        For each(SPListItem item in helpDeskList.Items) {
            Item.Update()
        }
    }
}

```

Figure 21: Pseudo-code for the *WorkflowItemsUpdateTimerJob*.

## 4.6 Administration

As mentioned in section 4.3.5, the only thing that differentiates the different roles in the Help Desk system is the permission level of the group which the role belongs. This is also the case for the Administrator. With Full Control permissions (meaning all permissions), the Administrator is responsible for creating the lists in the Help Desk Site, as well as populate the lists that lies beyond the permissions for the rest of the Help Desk team. These are, in fact, all lists except the *Help Desk List*, *Tasks List* and the *History Log List*.

This means that all list apart from these three just mentioned are managed only by the administrator. This is an important aspect for the security of the site; lists such as the *Client List* is best kept hidden from the rest of the Help Desk team, as it contains the client's password.



Hence, the main responsibility for the administrator is to manage these lists and populate them with new items. The administrator is also responsible for adding new users (or Help Desk team members) to appropriate groups.

## 4.7 Chapter Summary

Several programs are needed in order to setup the environment for the SharePoint application. These include a virtual machine, Windows Server 2003, Visual Studio 2005 with Workflow Extensions and WSS 3.0.

The system is divided into three parts: the *Ticket Registering* part, the *State Machine* part, and the *Administration* part. The *Ticket Registering* can be done in different ways: the ticket can be registered by a client from an external web site, it can be registered by a Telephone Operator, or it can be done by a dispatcher. The ticket can also be received by phone. The *Help Desk Support* team members are divided into groups depending on the role of the team member. These groups have an associated *permission level*, which states what the person can and cannot do in the SharePoint Site. As a member can have several roles, the member can belong to several groups as well.

The State Machine consists of three different activities: the *initialization*-, the *event driven*- and the *finalization* activity. In the *event driven* activity, all the *events* are implemented. There are two types of *events* that can be triggered: *OnWorkflowItemChanged* and *OnTaskChanged*.

## 5 Evaluation

---

*In this chapter, an evaluation of SharePoint Workflow will be presented. In particular, the State Machine workflow, since that is the workflow type used in the implementation. The benefits of working with workflows will be presented, as well as disadvantages and limitations. Also, issues such as performance will be brought up.*

---

### 5.1 Functional evaluation

#### 5.1.1 Benefits of using workflows

##### 5.1.1.1 Model creation

The main advantage with workflows (at least in WF) is that you're creating a model. This model might be some sort of UML diagram, or it might be a simple flow chart. In any case, these are models – a way to look at the process to be made. Workflows combines these two; when you are modeling your process, you're also building the application (again I'm only speaking of the WF, since it's what I've used implementing my workflow). Even if you made diagrams before you started implementing, it will be a pretty intuitive task transforming these into workflows. As an example of this, consider the similarities between figure 9 and figure 17. The first figure shows the diagram, made in StarUML<sup>5</sup>, which was made before implementing. The latter shows the same workflow, implemented in Visual Studio 2005. There are striking similarities. Hence, working with workflows becomes a more intuitive matter compared to other kinds of implementations with similar behavior.

---

<sup>5</sup> StarUML – A program used for designing diagrams (<http://staruml.sourceforge.net/>)

### **5.1.1.2 Prototyping**

Closely related to the previous benefit is the ability to quickly be able to set up a prototype, using the out-of-the-box features that come with the Visual Studio Extensions for Workflows. These enables you to, without coding, set up a fairly complex workflow in decent time.

### **5.1.1.3 Persistence**

A powerful tool which comes with the WF is persistence. WF provides for workflows to be stored and unloaded from memory in the middle of processing.

When running workflows reach a point in the code where they need to wait for something to happen, they are *serialized* (i.e. turned into binary) and *dehydrated* to the SharePoint database. By dehydration means that the workflow object is turned into a string and stored to the database. The benefit of this is that dehydrated workflows are no longer in memory, which is very useful when using workflows, because workflow tends to be a long process and they can go on for months. Also, they spend a lot of time idle, and during that time there's no need for them to be in memory. Later, when a workflow is deserialized and wakes up, or *rehydrates*, it continues where it left off in the code [13].

### **5.1.1.4 Rollback**

If any of the operations fail in a SharePoint Workflow, the process is rolled back to the last known good state. This is managed by the WSS, which establishes commit points between the workflow events, and in this way enables the workflow to rollback. This is good, because it decreases the amount of work you have to do when creating exception handlers for workflow activities [14].

### **5.1.1.5 No predetermined path**

This is the main advantage of using a State Machine Workflow instead of using a Sequential Workflow. Constructing the help desk system as a sequential workflow would surely result more difficult, as it would require a lot of while loops and if branching [11]. The reason why to use a state machine workflow in the help desk system is because of the amount of iterations it does: from a *New* ticket to a *Complete* ticket could easily result in five or more steps (such as sending a notification, reject or delegate ticket). The main reason why a State Machine is more suitable for these sorts of systems is because, although the number of steps increases,

the amount of code won't increase (at least not as much as with the sequential approach), since the logic is already provided in the different states.

## 5.1.2 Disadvantages

### 5.1.2.1 Persistence

This is not only one of the major advantages of using SharePoint Workflows (see 4.1.3), but also one of the major drawbacks, depending on how it's being used. As mentioned in chapter 4.5, objects are stored and *serialized*. This can however only be made if the object is *serializable*, meaning that it can be serialized. Many objects cannot be serialized, including some of the more important objects in WF (such as *SPList*<sup>6</sup> or *SPUser*<sup>7</sup>). This means that, after a dehydration/rehydration cycle, these objects will be invalid. So if you use a non-serializable object you have to store these object as an intermediate object, such as a string, and when you need to use it you have to make a local object and retrieve the adequate object (by using a getter or similar).

### 5.1.2.2 Custom features

The SharePoint platform is indeed really good and easy to use when it comes to create out-of-the-box workflows. However, there's a big leap from constructing an out-of-the-box workflow in SharePoint Designer, and making a custom workflows in Visual Studio. Disregarding the great amount of code which needs to be written, there are more things to have in mind when creating custom workflows, only when it comes to deploying it; you have to provide it with a Strong Name<sup>8</sup>, move dll-files, and make changes to xml-files. This is because there are so many components that come in play when deploying and integrating your workflow with SharePoint.

### 5.1.2.3 Loops

---

<sup>6</sup> SPList - Represents a list on a SharePoint Web site.

<sup>7</sup> SPUser - Represents a user in Microsoft Windows SharePoint Services.

<sup>8</sup> A strong name, or a Strong Key, is a naming convention used in computer programming. There can be more than one component (eg: DLL) with the same naming, but with different versions. This can lead to many conflicts. Strong key is a solution to maintain different versions of a component. A Strong Key (also called SN Key or Strong Name) is used in the Microsoft .NET framework to uniquely identify a component [20].

There is not possible to loop an activity or an event in a State Machine workflow. By loop I refer to a *while*, *do/while* or a *for* loop. When, for example, sending out multiple e-mails or tasks, this needs to be done *recursively*, i.e. the state must do a *self transition* (or by transition to an intermediate state, that transitions back to the earlier state).

## 5.2 Quantitative Evaluation

When it comes to measuring the performance of a State Machine Workflow, typical performance standards do not apply. This because the workflow engine is managing what is considered to be *long-running processes*, which means that the workflow may very well execute over the course of days, weeks, or months. Therefore, if it takes an extra nanosecond, or even some extra minutes for the process to move from one stage in the process to another is not very relevant [4].

What's relevant however is how well the workflow engine can manage processes simultaneously. Since the processes can span over such a long period, there are likely to be many processes running at the same time. A workflow engine must be able to keep these processes separated, and respond in reasonable time when any of these processes changes. It must also be able to accept a new process starting at any time without this affecting any other process.

For this reason, measuring transactions per second presents an incomplete picture of workflow performance.

### 5.2.1 Testing

Making a correct testing of the workflow engine is beyond the scope of this dissertation. The reason to this is because such tests would include running the workflow for weeks on several machines simultaneously. However, tests have been made by a company called *Mann Software* [4], in an 18 pages long article. The things that were measured were:

- The number of workflow starts per second.
- The number of tasks completed per second.
- The number of concurrent workflows.

The most important conclusion the authors of the article came to was that SharePoint workflows, when properly designed in a properly configured environment, can run *tens of thousands* of concurrent processes.

However, as the workflow they tested was the built-in *Approval* workflow, the test results should only be used as guidelines for the State Machine workflow implemented in the Help Desk system.

### 5.2.2 Limitations

Naturally, SharePoint and SharePoint Workflow have limitations when it comes to number of items, number of fields, number of concurrent users etc. An estimation of these limitations has been made by Microsoft [23]. A summary of the most important limitations is seen in table 4.

*Table 4: Limitations of SharePoint*

<b>Site Object</b>	<b>Guidelines for acceptable performance</b>	<b>Scope of impact when performance degrades</b>
<b>Web Site</b>	250, 000 per site collection	Site collection
<b>Document</b>	5 million per library	Library
<b>Item</b>	2,000 per view	List view
<b>Field type</b>	256 per list	List view
<b>Users in groups</b>	2 million per web site	Web Site

Most of the limitations in showed in table 4 are most certainly sufficient for the Help Desk: the number of web sites, documents, and users in groups in the Help Desk will probably only be a fraction of the limitations shown in table 4. However, the number of items in a list before performance degrades is not an unreachable number, especially when it comes to the number of *tasks* in the Help Desk. Since each item (i.e. errand) in the *Help Desk List* generates at least three tasks (and can theoretically generate an infinite amount of tasks), this limit of 2,000 items needs to be taken in consideration when, for instance, deciding the amount of time before items in a list needs to be stored in another place.



## 6 Conclusion

---

*This chapter will sum up the most important things in the project. Like in previous chapters, focus will lie on the State Machine Workflow. A general conclusion is also found, as well as a section describing possible future work with the project.*

---

### 6.1 General Summary

The Help Desk system can basically be divided into two parts: the SharePoint part (or the non-programming part), and the Workflow part (or the programming part).

The SharePoint part consists of a number of lists; lists where the entities of the system is stored, and lists where the relationships between those are stored. The most important list is the *Help Desk List*, where the errands, received from the clients, are registered. A registration can be made directly from this list, or from the client web page. In the latter case, it's the client that is making the registration; otherwise it's someone within the Help Desk team.

The Workflow part consists of a workflow, created in Visual Studio and deployed in SharePoint as a *feature*. This *feature* is associated with the *Help Desk List*, and each item created in this list automatically has an instance of the workflow associated with it. The workflow is responsible for each step in the lifetime of the new item, or errand (or *Ticket*), such as notifying the proper people when an errand has been registered, or keeping track of the time the errand has resided in each state, for statistical purposes.

### 6.2 State Machine Workflow conclusion

Using a State Machine Workflow when implementing a Help Desk system is in most cases a healthy approach. If compared with a manual Help Desk (without a system that handles tasks like notifications), a State Machine has obvious advantages: one thing is that it heavily



reduces the amount of internal e-mail that needs to be sent back and forth to the members of the Help Desk team. It also reduces the time it takes to distribute these e-mails, as they are composed automatically by the system. Making statistics and reports is also made easier, as each phase in the lifetime of a ticket is logged automatically by the system, along with necessary information.

A State Machine workflow is more complex than the other type of workflow provided in SharePoint: the Sequential Workflow. It is therefore important to evaluate the need of system before choosing which approach to take: for instance, how many states will the Help Desk need? If the system needs only a simple workflow, with two or three states, the best approach is most probably a Sequential Workflow. Otherwise, a State Machine is preferable.

An important quality for a good Help Desk system is the ability to store many concurrent items over a long period of time, and that during this period of time. The item should also be idle (not in memory) when not needed and only brought to memory when needed. Both of the workflow types share this ability.

According to a performance test (see chapter 5.2.1), a well designed workflow can run tens of thousands of concurrent workflow instances. Even though this most probably is a far greater number than the amount of tickets the Help Desk will have in progress at the same time, it still shows that workflows in SharePoint is a powerful tool worth taken in consideration, not only when dealing with Help Desk systems.

### **6.3 General Conclusion**

If returning to the initial question for this project, how SharePoint Workflows can facilitate the implementation of a help desk system, the quick answer to this question is: in many ways. The specific advantages of choosing a workflow can be found in chapter 5. To sum this up, one can say that the major advantage of workflows is the ability to with relatively few components be able to create a complex workflow, suitable for the specific needs of the company.

## 6.4 Project Summary

The main goal of this project: to implement and evaluate a Help Desk using a State Machine Workflow, was met, as well as the requirements of the system made by Xeratech. Not all the functionalities marked as optional by Xeratech has been met though: the FAQ has not been created. One reason for this is because the FAQ had a low priority from the start as it was considered almost outside the scope of this project.

The project itself was both instructive and interesting. SharePoint in general and Workflows in particular was a new area not only for me but also for the people at Xeratech, which resulted in a little less steep learning curve. Luckily, over the last couple of years, dozens of books, articles and other literature about SharePoint and Workflows have been written (very few of my references are more than three years old). This has been a great help.

The project has been made considering that the project eventually will be turned over and completely handled by Xeratech. Having this in mind, both the design and the implementation has been kept as simple as possible, without losing functionality.

Microsoft has invested large amounts of time and money to make SharePoint an intuitive and powerful tool, not only for programmers, but also for power users (non-programmers). As mentioned in chapter 4, the SharePoint site was made without writing a single line of code. So, making new lists and list items, sending tasks and e-mails to persons within the intranet has never been easier. SharePoint also makes it really easy for an admin (or a user with the proper permissions) to create permissions, groups and to manage user access.

However, when it comes to implementing a complex and robust workflow, SharePoint Designer just won't do the work. The SPD is suitable for doing simple workflows. So, when a more complex workflow is needed, Visual Studio is the only alternative (and a good one).

Numerous programs are required in order to work with SharePoint. This has resulted in almost as many new programs to learn and understand which has been a challenge at times. Creating and editing pages within SharePoint however, has been really intuitive and easy to do. The Workflow extension for Visual Studio 2005 has been a great tool when creating a workflow: it offers easy to understand drag-and-drop functions for most the components of a

workflow. Also, with the built-in activities that come with it, there has been no need to create custom activities. Visual Studio also provides good debugging that can be attached to the workflow instances in the SharePoint site.

## **6.5 Future work**

The Help Desk system could be extended in many ways. A full-feathered Help Desk system usually contains additional functionality such as the ability to search the database for already solved issues, a more sophisticated priority system, or a more detailed web site for clients, with more functionality. Most of these things could without too much difficulty be implemented in SharePoint (SharePoint already have extensive support for searching sites and lists within the intranet).

The State Machine Workflow can also be further developed. For instance, the current version of the workflow contains four types of notifications. This number can increase with, for example, a notification to the client when a major change has been made to any of his or her errands, or a more complex hierarchy of support members, where these roles needs to be specially notified whenever certain criteria is met.

## References

- [1] Xeratech AB. (2009) *Detta är Xeratech*. <http://www.xeratech.se/>
- [2] Microsoft. (2009). *Create a custom list form*.  
<http://office.microsoft.com/en-us/sharepointdesigner/HA101191111033.aspx>
- [3] Webb, J. (2008) *Essential SharePoint 2007, Second Edition: A Practical Guide for Users, Administrators and Developers*. O'Reilly.
- [4] Microsoft. (2009). *Workflow Scalability and Performance in Windows SharePoint Services 3.0*. <http://msdn.microsoft.com/en-us/library/dd441390.aspx>
- [5] Wayne , W.; Tynes, W. ; Cathey S. (2007) *Microsoft® SharePoint® Server 2007 Bible*. John Wiley & Sons.
- [6] Sterling, D. M. (2008). *Microsoft® Office SharePoint® Server 2007: The Complete Reference*. The McGraw-Hill Companies.
- [7] Microsoft. (2009). *What is Microsoft Office SharePoint Server?*  
<http://www.microsoft.com/sharepoint/prodinfo/what.mspx>
- [8] Wikipedia. (2009). *WYSIWYG*. <http://en.wikipedia.org/wiki/WYSIWYG>
- [9] Chappell , D. (2006) *Understanding Workflow in Windows SharePoint Services and the 2007 Office System*. Chappell & Associates.
- [10] Mann, D. (2007). *Workflow in the 2007 Microsoft Office System*. Apress.
- [11] Myers, B. R. (2007). *Foundations of WF: An Introduction to Windows Workflow Foundation*. Apress.
- [12] Wikipedia. (2009). *State Machine*. [http://en.wikipedia.org/wiki/State\\_Machine](http://en.wikipedia.org/wiki/State_Machine)
- [13] Hao, E. (2006). *So You Want to Develop SharePoint Workflows...: Key Concepts for Understanding How to Develop SharePoint Workflows in Visual Studio*. Microsoft Corporation.
- [14] Microsoft. (2009). *How Windows SharePoint Services Processes Workflow Activities*.  
<http://msdn.microsoft.com/en-us/library/ms442249.aspx>
- [15] Microsoft. (2009). *.NET Framework Developer's Guide: Strong-Named Assemblies*  
[http://msdn.microsoft.com/en-us/library/wd40t7ad\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/wd40t7ad(VS.71).aspx)

- [16] UC4 Software. (2008). *Workflow vs. Application Automation Tools: Choosing the Right Tool for the Job*. White paper. <http://www.uc4.com/en/about-uc4/resources/uc4-white-papers/>
- [17] Merriam-Webster (2009). *Automation*.  
<http://www.merriam-webster.com/dictionary/automation>
- [18] Wikipedia. (2008). *Orchestration*.  
[http://en.wikipedia.org/wiki/Orchestration\\_\(computers\)](http://en.wikipedia.org/wiki/Orchestration_(computers))
- [19] Bender, J. (2008). *Workflows vs. Orchestrations*.  
<http://nplus1.org/articles/workflows-vs-orchestrations/>
- [20] Wikipedia. (2008). *Strong Name*. [http://en.wikipedia.org/wiki/Strong\\_name](http://en.wikipedia.org/wiki/Strong_name)
- [21] Wikipedia. (2009). *Ticket*. [http://en.wikipedia.org/wiki/Ticket\\_\(IT\\_support\)](http://en.wikipedia.org/wiki/Ticket_(IT_support))
- [22] Wikipedia. (2009). *Service Level Agreement*.  
[http://en.wikipedia.org/wiki/Service\\_level\\_agreement](http://en.wikipedia.org/wiki/Service_level_agreement)
- [23] Office IT and Servers User Assistance. (2009). *Planning and architecture for Office SharePoint Server 2007, part 2*. Microsoft Corporation.

# A Help Desk Site

## A.1 Team Site Template



The *Team Site Template* is one of the templates in the *Collaboration* group, which contains site templates designed to help teams within an organization to work on projects, collaborate on documents, or share information. It's one of the four categories of templates shipped with SharePoint, adding up to a total of 40 templates. This template is used throughout the whole site, not only the *Help Desk* site.

## A.2 Assigned To Rejected Notification Form

The dialog box has a title bar with 'Attach File', 'Delete Item', and 'Spelling...'. It contains two main sections: 'New Assigned To' and 'Comments'. The 'New Assigned To' section has a text input field containing 'Sven Svensson' and a dropdown arrow, with the instruction 'Select a new Assigned To from the errand consultant group.' below it. The 'Comments' section has a rich text editor toolbar and a text area containing 'Bengt Bengtsson wrote: Handle this errand.' Below the text area is the instruction 'Add additional comments that will be sent with the verification.' At the bottom left, it shows 'Version: 2.0', 'Created at 2009-04-20 10:27 by Fanny Stensson', and 'Last modified at 2009-04-20 10:39 by Sven Svensson'. At the bottom right are 'OK' and 'Cancel' buttons.

As seen in figure A.2, this form is a little different compared to the other notification forms, in the sense that it cannot be replied to, and hence has no such buttons. The only thing the receiver can do is to assign the errand to another person.

## A.3 Delegated To Affirmation Form

The dialog box has a title bar with 'Attach File', 'Delete Item', 'Spelling...', and '\* indicates a required field'. It contains two main sections: 'Delegation Response \*' and 'Comments'. The 'Delegation Response \*' section has two radio buttons: 'Accept' (selected) and 'Reject', with the instruction 'Select the response to the delegation.' below them. The 'Comments' section has a rich text editor toolbar and a text area containing 'Ok, I'll take it.' Below the text area is the instruction 'Add additional comments that will be sent with the verification.' At the bottom left, it shows 'Version: 2.0', 'Created at 2009-04-21 11:26 by MOWIT\Administrator', and 'Last modified at 2009-04-21 11:26 by MOWIT\Administrator'. At the bottom right are 'OK' and 'Cancel' buttons.

In this form, the receiver has the option to Accept or to Reject the delegation.

## A.4 Solved Verification Form

The screenshot shows a software window titled "Solved Verification Form". At the top, there is a toolbar with icons for "Attach File", "Delete Item", and "Spelling...". A note on the right states "\* indicates a required field".

The main content area is divided into two sections:

- Solved Response \***: This section contains two radio buttons. The first is selected and labeled "Verify (sets errand to Completed)". The second is labeled "Reject (sends back errand to Assigned To)". Below these is the instruction: "Select the response to whether the errand is completed or not."
- Comments**: This section features a rich text editor toolbar with icons for bold, italic, underline, list, and other text formatting options. Below the toolbar is a text input field containing the text "Good job with that errand!". Below the input field is the instruction: "Add additional comments that will be sent with the verification."

At the bottom left, the version information is displayed: "Version: 1.0", "Created at 2009-04-20 10:49 by Fanny Stensson", and "Last modified at 2009-04-20 10:49 by System Account". At the bottom right, there are "OK" and "Cancel" buttons.

Basically, this is the same form as the Delegated To Affirmation Form, the only difference is the text of the “Yes or No” radio buttons. This is needed to prevent ambiguity for users.



## B Permissions and Groups

### B.1 Help Desk Permissions and Groups

Permissions: Help Desk Site

Use this page to assign permission levels to users and groups. This Web site does not inherit permissions from its parent.

New ▾ | Actions ▾ | Settings ▾

<input type="checkbox"/>	Users/Groups	Type	User Name	Permissions
<input type="checkbox"/>	Administrator Group	SharePoint Group	Administrator Group	Full Control
<input type="checkbox"/>	Dispatcher Group	SharePoint Group	Dispatcher Group	Errand Consultant Permission Level
<input type="checkbox"/>	Errand Consultant Group	SharePoint Group	Errand Consultant Group	Errand Consultant Permission Level
<input type="checkbox"/>	System Account	User	SHAREPOINT\system	Limited Access
<input type="checkbox"/>	Telephone Operator Group	SharePoint Group	Telephone Operator Group	Telephone Operator Permission Level
<input type="checkbox"/>	Verifier Group	SharePoint Group	Verifier Group	Verifier Permission Level

## B.2 Help Desk Permissions Levels

Name:

Description:

---

Select the permissions to include in this permission level.

**Select All**

**List Permissions**

- Manage Lists** - Create and delete lists, add or remove columns in a list, and add or remove public views of a list.
- Override Check Out** - Discard or check in a document which is checked out to another user.
- Add Items** - Add items to lists, add documents to document libraries, and add Web discussion comments.
- Edit Items** - Edit items in lists, edit documents in document libraries, edit Web discussion comments in documents, and customize Web Part Pages in document libraries.
- Delete Items** - Delete items from a list, documents from a document library, and Web discussion comments in documents.
- View Items** - View items in lists, documents in document libraries, and view Web discussion comments.
- Approve Items** - Approve a minor version of a list item or document.
- Open Items** - View the source of documents with server-side file handlers.
- View Versions** - View past versions of a list item or document.
- Delete Versions** - Delete past versions of a list item or document.
- Create Alerts** - Create e-mail alerts.
- View Application Pages** - View forms, views, and application pages. Enumerate lists.

**Site Permissions**

- Manage Permissions** - Create and change permission levels on the Web site and assign permissions to users and groups.

- View Usage Data** - View reports on Web site usage.
- Create Subsites** - Create subsites such as team sites, Meeting Workspace sites, and Document Workspace sites.
- Manage Web Site** - Grants the ability to perform all administration tasks for the Web site as well as manage content.
- Add and Customize Pages** - Add, change, or delete HTML pages or Web Part Pages, and edit the Web site using a Windows SharePoint Services-compatible editor.
- Apply Themes and Borders** - Apply a theme or borders to the entire Web site.
- Apply Style Sheets** - Apply a style sheet (.CSS file) to the Web site.
- Create Groups** - Create a group of users that can be used anywhere within the site collection.
- Browse Directories** - Enumerate files and folders in a Web site using SharePoint Designer and Web DAV interfaces.
- View Pages** - View pages in a Web site.
- Enumerate Permissions** - Enumerate permissions on the Web site, list, folder, document, or list item.
- Browse User Information** - View information about users of the Web site.
- Manage Alerts** - Manage alerts for all users of the Web site.
- Use Remote Interfaces** - Use SOAP, Web DAV, or SharePoint Designer interfaces to access the Web site.
- Use Client Integration Features** - Use features which launch client applications. Without this permission, users will have to work on documents locally and upload their changes.
- Open** - Allows users to open a Web site, list, or folder in order to access items inside that container.
- Edit Personal User Information** - Allows a user to change his or her own user information, such as adding a picture.

**Personal Permissions**

- Manage Personal Views** - Create, change, and delete personal views of lists.
- Add/Remove Personal Web Parts** - Add or remove personal Web Parts on a Web Part Page.
- Update Personal Web Parts** - Update Web Parts to display personalized information.

