Abstract— This Masters Project is one component of a distributed Workflow Management System called WorkTrak. This product currently is a distributed application that features a Java Swing Client communicating with a Java Server using a Remote Method Invocation (RMI) Server that stores the state of the system. This project explores the benefits of a .Net web application client for the Workflow Process Management. The aim of this project is to design and implement a web based solution for the Process Manager User Interface of the WorkTrak Workflow Management System.

The Process Management Console is a web application developed using ASP.NET and C#. It is meant to provide an ‘owner’s view’ of workflow processes being enacted and to enable the user to view and change the current status and other properties of an instantiated process and its activities. The Console allows the user to also view a list of the current process definitions and activities and to instantiate a new process. The user is able to also view and modify the input and output products defined for the instantiated activities or processes. This web application replaces the Process sub module in the existing Java Swing Client.

I. INTRODUCTION

A. Background

The Workflow Management Coalition (WfMC) defines a workflow as “The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules”[1]. A workflow is normally a series of activities that are to be performed in a pre-arranged order. These activities can be assigned to users or roles. The activities may have input and output products that also form a part of the workflow. This necessitates the use of a framework to define, manage and track workflows and the associated activities. Workflow Management System is defined by the WfMC as “A system that defines, creates and manages the execution of work-flows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications” [1]. This provides the definer of the workflow also known as the ‘owner’ with a process measurement tool that can be used to improve business practices.

B. Objective

The WorkTrak workflow management system called WorkTrak is being developed as an evolving research product [3,18,20]. It is based on the Workflow Reference Model [2] published by the WfMC and provides the user with the ability to define and enact workflows in accordance with the published standards. The current WorkTrak system is a java-based client server application. This project aims to convert the workflow process manager tool of the existing application to a web – based application using .Net technology.

The interface allows users to view the current workflow definitions and instantiate a particular workflow. The user is able to assign the activities defined in the workflow to particular resources, view the status of the process and its activities and the products that are obtained as a result.

C. Scope of the project

Developing a client for the WorkTrak Workflow Management System is a collaborative effort by four students[22,23,24]. Each person had a specified module to develop with the three client modules being dependent on the Java Web Service for testing and usability.

This project is concerned with the analysis, design, implementation and testing of the process manager tool, which is one of the modules of the WorkTrak System client as a .Net based web application.

The plan is to use the existing RMI WorkTrak Server via a Java web service that can be accessed by the Web based Client. The project provides the interface to enact and manage workflows, assign activities to particular roles, and view the status of the activities and the workflow as a whole.

The Java Web Service is being developed by Sahni [23] while the Worklist Handler and Definition tools are being developed by Sundaram[24] and Liu[22] respectively.

D. Approach

The development of this project follows the software engineering life cycle. The waterfall model is adopted beginning with the requirements gathering phase followed by the design, coding and testing phases. Since the client is an Asp.net web application, a java web service is necessary to act as a link between the client and the Java RMI WorkTrak server. Importance is given to the design of the client user interface to be as user friendly as possible. The tool is a part of a web based application that acts as the user interface for the
WorkTrak Workflow Management System. The development platform is Visual Studio.Net and the web application is hosted on Microsoft Integrated Information Systems (IIS) server or Cassini web server. Tomcat and Axis are used to develop the Java web service that acts as a link between the WorkTrak client and RMI WorkTrak server.

II. REVIEW OF LITERATURE

The working of an organization is highly streamlined by setting procedures and standards in place to perform the operations required for its proper functioning. These procedures are processes that consist of ordered activities, their input and output products and the various users involved. A workflow can be generated by defining any of these processes in terms of the activities, the inputs fed into each activity and the output obtained from it and the users that perform or oversee the performance of the activities. The activities may have a predefined set of input and/or output products and instructions by which the activity needs to be performed. Advances in technology have facilitated the automation of workflow processes, where the output product of one activity can be automatically routed as the input product of the next activity. Certain activities themselves can be completely automated to increase the productivity and efficiency of the business process. There arises a need for Workflow Management Systems (WFMS) that can “define, manage, monitor and execute workflows through the execution of software whose order of execution is driven by a computer representation of the workflow logic” [2]. Figure 1 shows a conceptual diagram of the use of workflow management systems.

A. Workflow Management Systems

Workflow Management Systems (WFMS) may be implemented in a variety of ways depending on the operating environment and the complexities of the business processes involved. However, since Workflow Management Systems are based on similar system frameworks, there are common features that allow interoperability and integration between different products.

On a high level, Workflow Management Systems provide support in the following functional areas:
1) **Build-time function**: concerned with defining, and possibly modeling, the workflow process and its constituent activities [2].
2) **Run-time control functions** concerned with managing the workflow processes in an operational environment and sequencing the various activities to be handled as part of each process [2].
3) **Run-time interactions** with human users and IT application tools for processing the various activity steps [2].
4) Some combination of process administration, history, analysis and audit [3].

Figure 2 shows the characteristics of a generic Workflow Management System.

![Figure 2. Generic Workflow Management System Characteristics](#)
The roles of these components are described below:

1) **Process Definition Tool**: This tool is used to define a process. The definition of a process may include defining the activities within the process, the sequence of the activities, the input and output products of each activity, instructions to perform each activity and the role in which the participant performs each activity. [2]

2) **Workflow Enactment Service**: This tool allows the users to initiate a particular workflow definition. The user may ‘own’ the initiated workflow and be able to monitor its status, assign activities to the participants in the workflow and view the input and output products. [2]

3) **WorkList Handler**: This tool provides the users with a list of activities that are assigned to them. The users can view assigned activities, modify product bindings and update the status of activities assigned to them. [2]

**B. Benefits of Workflow**

1) **Improved Productivity**
   “The automation of business processes invariably results in the elimination of many unnecessary steps and a reduction in the time spent processing necessary steps”. [5]

2) **Better Process Control**
   “This leads to improved management of business processes achieved through standardizing working methods, monitoring work in progress and the availability of audit trails provide compliance and control”. [5]

3) **Improved Customer Service**
   “Consistency in processes leads to greater predictability in levels of response to customers and reduced response times”. [5]

4) **Business Process Improvement**
   “Focus on business processes leads to their streamlining and simplification as information on process efficiency is generated from the workflow system”. [5]

5) **Application Integration**
   “Integration to legacy systems from a single case management application provides the control point for integrating disparate processing systems”. [5]

**C. Workflow Enactment Service**

Predefined process definitions can be interpreted and instantiated by the workflow enactment service. It controls the sequencing of activities, adding work items to the user work lists and invoking application tools as necessary.

One of the distinguishing features of the workflow infrastructure is its ability to distribute tasks and information between participants. The distribution of the functions can be at different levels: within a workgroup, within a department or within an organization. Also, a variety of communication mechanisms like email, messaging, file sharing etc. can be used depending upon the scope of the communication. A diagram of the interaction within the workflow enactment service is shown in figure 4.

**D. WorkTrak Workflow Management System**

The WorkTrak Workflow Management System is based on the Workflow Reference Model defined by the Workflow Management Coalition (WfMC), as well as evolving workflow architectures[3,18,20]. It is a distributed application that consists of a WorkTrak Server based on Java Remote Method Invocation (RMI). [18] The user can access the server functionality through the Java Swing Client. The state of the server can be saved and retrieved from an Xml file by using the save and restore functions. A high level component
The diagram of the WorkTrak Workflow Management System is shown in figure 5.

![Diagram of WorkTrak Workflow Management System](image)

**Figure 5. Original WorkTrak High Level Component Diagram**

The WorkTrak Server provides the functionality to define workflow processes, instantiate them and assign tasks to users. The users have the provision to view the activities assigned to them and change the status and product bindings of the activities. The Workflow Client contains a JTabbed Pane with each panel providing core functionality. A screenshot of the client is shown in figure 6.

![WorkTrak Java Swing Client](image)

**Figure 6. WorkTrak Java Swing Client**

The Definition Tab provides details about the workflow definitions and allows the user to instantiate a process. The process tab is modeled on the enactment service. It allows the user to have an ‘owner’ s view of the workflow, view the status of the ongoing processes and activities, assign activities to users and change the state of the processes and activities. It also provides a view of the history of each of the instantiated processes. The activity tab is the worklist handler component where the users can view the activities assigned to them, change product bindings and states.

However, “WorkTrak differs from traditional workflow due to its design that supports the tracking of individual contributions to coordinated group activities” [18].

The aim of this project is to develop a web based client that performs the functions of the current Java Swing Client and explores the facilities available and benefits of .Net Web Applications.

**E. Asp.Net Web Application**

Web Applications are defined as ‘A software program that uses HTTP for its core communication protocol and delivers Web-based information to the user in the HTML language [7]. Web Applications need to be deployed on a web server like Internet Information Systems (IIS), Cassini, and Apache Web Server etc. Dynamic Web-Applications typically have some sort of scripting on either the client or server side. ASP.net is the next generation of Microsoft's Active Server Page (ASP), a feature of their Internet Information Server (IIS). ASP.net is different from ASP in two major ways: it supports code written in compiled languages like C#, C++, Visual Basic, and Perl, and, it features server controls that can separate the code from the content, allowing WYSIWYG editing of pages [8].

![ASP.Net Sequence Diagram](image)

**Figure 7. ASP.Net Sequence Diagram [19]**

Figure 7 shows the sequence of requests that happen when an ASP.Net (.aspx) page is loaded onto a web browser and whose code behind uses an Asp.Net Web Service. The HTTP request is first send from the client or browser to IIS that has deployed the Web Application. The request is interpreted by the ASP.NET framework running in the IIS. It gets the .aspx file, translates it into HTML and sends it to be displayed on the web browser or client. The user then performs the necessary actions on the webpage and may click the submit button. This routes another request to IIS and the ASP.Net framework to get the C# codebehind for the .aspx file and call the necessary methods. The diagram also shows how the client interacts with a web service through the .asmx file and its codebehind library.

**F. Why use ASP.Net Web Applications?**

Along with the advantages of easy maintenance and accessibility, offered by web applications, ASP.Net also has some potential advantages other web development models such as:

1) **Enhanced Performance**

The code is compiled in the .Net Framework Common Language Runtime (CLR). This allows the ASP.Net to take can take advantage of early binding, just-in-time compilation and other such services provided by the .Net framework.[11],[12]

2) **State Management**

ASP.NET provides application-state and session-state management, which allows managing states of Web applications. [12], [13]
3) Enhanced Security

ASP.NET provides different levels of security (windows, forms etc.) to authenticate and authorize users. [11], [13]

4) Flexibility

Because ASP.Net is based on the Common Language Runtime of the .Net Framework, the features of the platform like Base Class Libraries, Messaging etc. are available to it. The codebehind files can be written in any of the .Net languages and are inter-operable. [12]

III. METHODOLOGY

The ASP.Net Process Manager Web Application is developed using the traditional Waterfall Model. The requirements were gathered through meetings with the professors who are to be the users and also through a thorough study of the existing WorkTrak Workflow Management System.

A. High Level Architecture of WorkTrak

As mentioned earlier, the WorkTrak system constituted a Java RMI WorkTrak Server that communicated with a Java Swing WorkTrak Client. It is agreed by the development team that the current WorkTrak version use the existing RMI WorkTrak Server. The three core workflow components, Definition Process Manager, and Worklist Handler, were to be separate applications that provided the core functionality. In addition, the team decided that each of these modules is to be an ASP.Net Web Application with a C# Codebehind. The Definition Tool is being developed by Liu and is estimated to be completed by September 2004 [22]. It provides the interface for adding and modifying workflow definitions and activities, and tools to add, modify and remove other workflow components like users, roles and products. The WorkList Handler Tool is developed by Sundaram [24]. It provides the user with a list of activities assigned to them and their due dates. The clients connect to the Java RMI WorkTrak Server through a Java Web Service that acts as a link between the Server and the C# client. A high level architecture diagram of the system is show in figure 8.

This paper is concerned with the development and implementation of the Process Management Client Application.

B. Design

The Process Manager Application consists of two main views and a login Screen

1) Add Process View

This view allows the user to view the list of current definitions and instantiate a process based on a particular definition. The name of the process needs to be provided by the user and the user has the option to also provide the names of the products generated by the process.

2) Process Owners View

This view lists the processes, sub-processes and activities instantiated by the current user. It also provides an interface for the owner to view the status of an instantiated process and its activities and sub-processes. The Owner is able to assign an activity to a particular participant, change the status of the activity or process and change or view the product bindings for the activity or process. The Process owners View also provides the user with a view of the history of a particular process.

3) Login Screen

A login screen is also provided for filtering purposes. This screen provides a list of all valid usernames and allows the user to select the desired name. This username is set as the owner name for adding a new process in the Add Process View and is also used to get a list of the processes instantiated by the user for the Process Owner View.

C. Implementation

As mentioned earlier, the Process Manager is developed as an ASP.Net application using a C# codebehind. The ASP.Net (.aspx) web pages were developed using Visual Studio .Net IDE. The functionality in C# code behind these .aspx pages were added in using text editors like VIM [9], Notepad and EditPlus 2[10].

1) User Defined Controls

Two user defined controls: Banner and TabPanel[21] were developed to provide a better look to the client and to improve the usability. The Banner control displays the name of the client and a logo. It helps to retain the same look across different aspx web pages. The TabPanel control provides tab page functionality to the client. This increases the ease of navigation of the webpage and hence increases usability.

2) Third Party Controls

The nature of information displayed is hierarchical where one process definition can contain multiple activities and sub-processes. Each of these sub processes can in turn contain multiple activities and sub processes and so on. Such data is probably very well represented as a tree with nodes and sub-nodes. A third party developed Software, TreeLink by DevMansion [9] is used to represent the list of definitions in

![Figure 8. Architecture of WorkTrak Workflow Management System](image-url)
the add process view and the list of processes in the process owners view.

3) ASP.Net Web pages

The functionality described in the design section above is implemented as a series of interlinked webpages.

1) Login Page

The Login Screen is a single aspx page that allows the user to select the desired user name. It connects to the WFAgentWS web service to all the user names present in the server and displays it in a drop-down list box. The user selects the desired name and clicks the submit button to be logged on with that user name. Figure 9 shows a screen shot of the login Screen.

2) Add Process Page

The Add Process View is implemented as a webpage with three frames. The top frame displays the banner and the tabPanel. The left pane displays the treeview with the list of definitions and the activity and sub-process defined as a part of each process definition. It connects to the WFProcessDefWS web service to get this information from the server. The right frame or the ‘workspace’ displays the controls that allow the user to add a process. Figure 10 shows the layout of the add Process pane.

This page connects to the WFProcessDefWS web service to get the products for the selected definition. Then it calls the WFProcessWS web service to instantiate a process according to the selected definition.

3) Process Page

The process page is the implementation of the Process Owner’s view. It is also made up of three frames similar to the add process page. The top pane contains the banner and tabpanel control to keep the look uniform through out the application. The left frame connects to the WFProcessWS web service to get the list of processes and their sub enactments. This information is displayed in a clickable treeview. The right pane displays the information about the selected process or activity.

If the user selects a process or sub process on the treeview, the process details page is displayed. It contains the name of the Process, its Owner and Status of the process. The user can change the owner and the status of the process and click update to change the properties of the process. The process details page also shows a tabular history of state changes of the process over time as shown if figure 11. A product and product binding is also displayed. The user has the ability to update the product bindings or delete the product if so desired.

If the user selects a process or sub process on the treeview, the process details page is displayed. It contains the name of the Process, its Owner and Status of the process. The user can change the owner and the status of the process and click update to change the properties of the process. The process details page also shows a tabular history of state changes of the process over time as shown if figure 11. A product and product binding is also displayed. The user has the ability to update the product bindings or delete the product if so desired.
This page connects to the WFProcessWS to get and update the information about the process.

If the user selects an activity on the left hand tree view, the activity details are displayed on the right hand side as shown in figure 12. The activity details page displays the name of the activity selected. It also displays the name of the user that the activity is assigned to and the status of the activity in dropdown listboxes. These values can be changed and updated to the server using the update button. Directions for performing the activity are also displayed.

![Figure 12. Process Page with Process Details](image)

As in the product details page, a list of products and their bindings belonging to the activity are displayed. The product bindings can be edited or the products can be deleted. This page connects to the WFActivityWS to get and set these values.

D. Deployment and Testing


The client is tested on functionality by creating processes and also by modifying the properties of the instantiated processes and activities. The look and feel is tested by changing the font size of the browser and the resolution of the testing desktop. Window sizes were also modified to check their effect on the application. Finally the client is tested on multiple browsers to check its compatibility.

IV. RESULTS

After debugging the Client and the Java Web Service, the client could successfully connect to the RMI server. The expected information is obtained from the server for displaying the list of users in the login page, the process and definition trees and the details of the process and activity selected. The values that are modified are successfully updated to the server and redisplayed on the client. Validation of the input values however is not implemented at this time.

The client maintained its look and feel even if the size of the font or the resolution of the user’s environment changed. It displays correctly on multiple browsers such as Mozilla and Netscape Navigator. However, because of the nature of ASP.Net controls, however, it is best viewed in Internet Explorer 5.0 or higher.

As mentioned earlier, the project to develop a client for the WorkTrak Workflow Management System is a collaborative effort. The Process Manager Module is dependent on the completion of the Java Web Service for in-depth testing and usability. Because of such environmental and time constraints, extensive stress testing and usability testing is not carried out.

V. CONCLUSION AND FUTURE WORK

A Process Management Console is developed to provide a web based interface for a workflow management system. The user can connect to the system with a user name and be able to monitor the status of process and activity enactments. User Authentication and user input validation however is not addressed in this project and would be a good enhancement to the client.

ACKNOWLEDGMENT

Krithiga Priyadarshini would like to thank Dr. Timothy E. Lindquist for the help and support given at the meetings. Sincere thanks are also extended to Professor Richard O. Whitehouse for his feedback on the User Interface Design, which helped her develop the application from the end users point of view. She would also like to thank the rest of the development team for their input.

REFERENCES

[23] Sahni, R., Developing Java Web Services to Expose the WorkTrak RMI Server to the Web and XML-Based Clients, Masters Project for Department of Computing Studies and Technology, Available HTTP: http://pooh.east.asu.edu/Lindquist/WorkTrak