Building Virtual Warehouses using Workflows and the Web

Krithi Ramamritham and *Mohan Kamath*
Department of Computer Science
University of Massachusetts
Amherst MA 01003
{*krithi,kamath*}@*cs.umass.edu*

Abstract

Whereas the Web has been growing at a rapid rate to become a global information source, Workflow management systems (WFMSs) are gaining popularity as a tool to automate business processes. The number of companies adopting the Web as a platform for conducting their daily business activities and for electronic commerce is expected to sharply increase. Hence, there is a need for developing WFMSs that address the requirements of conducting business over the Web and making efficient use of Web technology.

In this paper, we discuss how the technology developed for workflows can be applied to meet the needs of electronic commerce applications over the Web. We also show how the same technology can be used to build virtual warehouses from information sources available on the Web. As an example of this we show how the various types of information needed for intelligent decision making by an investor in stocks can be automatically gathered and presented to a user using workflow techniques.

Keywords: Workflow Management, Correctness, Failure Handling, Coordinated Execution, Transactional Workflows, Data warehouses, World Wide Web.

† Supported by NSF grant IRI-9314376.

1 Introduction

Because workflow management systems (WFMSs) coordinate the execution of programs/tasks distributed across an enterprise, they are gaining popularity as a tool to automate business processes. Simultaneously, the Web has been growing at a rapid rate to become a global information source. The number of companies adopting the Web as a platform for conducting their daily business activities and performing electronic commerce is expected to sharply increase. Hence, there is a need for developing WFMSs that address the requirements of conducting business over the Web and making efficient use of Web technology.

Information available on the Internet and the World Wide Web can be viewed as global virtual databases. When one is interested in data pertaining to a specific domain, e.g., stock related information, these form virtual data warehouses. Given the autonomy of many of these data sources, as well as the temporal nature of some of the data, it may not be possible to materialize these warehouses. Hence depending on the specific nature of a user's query, the system must be able to develop a plan by which to access the different components of the virtual warehouse and provide a timely response. Specifically, the process of gathering data from distributed sources has to be done efficiently and correctly. Applying workflow ideas, we have developed a system that is be able to gather needed information *without a user explicitly asking for each and every piece of relevant information*. The collection and collation of relevant information is expressed via workflow specifications and the system automatically accesses the necessary sources and responds in a timely fashion to the needs of the users. With such information in hand users can conduct electronic commerce, by executing transactions over remote sources, again using workflow and web technology. In the rest of the paper, after introducing workflows, we discuss how this can be accomplished.

2 Workflow Management Systems

As described by the Workflow Management Coalition (WfMC) [Hol94], a WFMS consists mainly of a workflow engine and some application agents. In addition, there are other components for modeling, administration and monitoring. The workflow engine and the tools communicate with a workflow database to store and update workflow relevant data. The components are connected as shown in Figure 1. A workflow schema (workflow definition) is provided by a workflow designer with the aid of the modeling tool. Later, based on the same workflow schema, several instances of the workflow can be created. A workflow schema is essentially a directed graph with nodes representing the steps/tasks to be performed. The arcs connecting the steps are of two types: data and control arcs. Each data arc specifies the mapping of the output data from one step to the input of the other. Each control arc indicates the ordering sequence between two steps. It may also have a condition associated with it to indicate that the succeeding step is to be executed only when the condition evaluates to true. As described in the schema, the step is provided with some input by the workflow engine and the output of the step is returned to the workflow engine. The individual steps can be executed automatically or manually or both. A manual step consists of an activity such as filling a form or making a phone call based on the data provided to a human. An automatic step corresponds to the execution of a specified program at a particular agent. The workflow engine communicates with the agent to

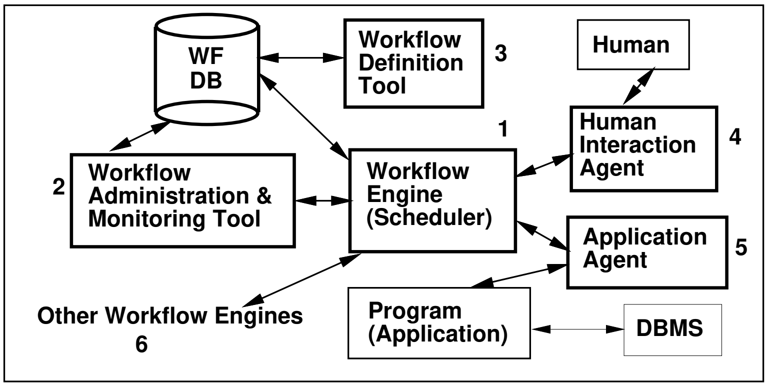

Figure 1: *Traditional Workflow Management System*

achieve this. A step can also be a combination of the two, $i.e.$, a human supervising the execution of a program at an agent. Staff members have to be assigned to work on steps that have to be executed by humans. Hence, in addition to defining workflow schemas, the modeling tool also facilitates the description of the organizational structure consisting of the staff hierarchy, staff name with their designation and so forth. Staff members interact with the WFMS through a 'to-do' list that indicates which of the steps (from various executing workflow instances) should be executed by that staff member. The 'to-do' list is presented to a staff member when the staff logs into the system using a workflow client. With the help of the workflow database (WFDB), most WFMSs provide forward recovery to handle the failure of the workflow engine, $i.e.$, each workflow is continued from the state it was in before the failure occurred rather than starting from the first step.

3 Workflows over the Web

In order to exploit the potential of the Web, companies offering workflow products now provide a *Web interface* to WFMSs [Act, Flo]. This allows both the employees and the customers of a company to interact with the WFMS through the Web. Employees can access their 'to-do' list (a list of work items an employee, more generally, a processing entity, is responsible for.) via a Web browser instead of using a special client as in traditional WFMSs. Thus employees can interact with the WFMS wherever they are located as long as they have access to a Web browser. Also, when a customer submits an order via a Web form, an appropriate workflow is automatically triggered to handle the customer order. However these WFMSs lack facilities for workflows whose steps directly access and update data from Web servers. In short, they support customer-to-business transactions over the Web but not business-to-business transactions. The latter is necessary to support workflows whose steps are business transactions executed by Web servers. Once we have such facilities, as we will show, workflows can be used for creating global data warehouses and for conducting electronic commerce.

It would be more beneficial to provide the WFMS with additional knowledge about the Web, its protocols and data format such that workflows can directly access information by interacting with Web servers. This will allow companies to automate their business processes on the Web by allowing steps to automatically query/update

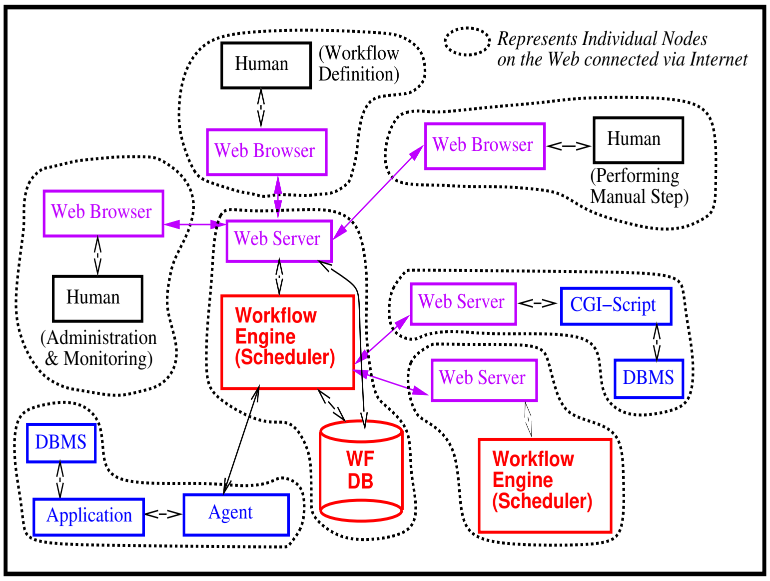

Figure 2: *Workflow Management over the Web*

information from Web servers located both within and outside the organization. The additional knowledge required is provided by the workflow system developer and the workflow designer. Combining workflow management and Web technologies in this manner will offer immense benefits for conducting business over the Web. In addition to customer-to-business transactions, workflows consisting of business-to-business transactions can be executed over the web. For example, it will be possible to implement Web workflows that handle supply chain management where a group of suppliers and customers distributed around the world cooperate as trading partners (See Section 5). The formation of such a *virtual enterprise* would make it possible to implement JIT (Just In Time) method of inventory control to save money on inventory, warehousing and handling costs. Thus a company manufacturing a product can use Web workflows to place orders for the individual components needed to make the product immediately after it receives an order for the product. Steps of Web workflows will use the same Web interface provided by the supplier for its customers. To make Web workflows feasible, it is necessary to develop WFMSs that address the special requirements of conducting business over the Web and making efficient use of Web technology.

Now, we discuss how a WFMS for the Web differs from a traditional WFMS. The Web is essentially a client-server based distributed system where browsers acting as clients communicate with Web servers using the HTTP protocol [HTT]. When a workflow is executed on the Web, the Web servers play the role of agents. The CGI-scripts at the Web servers are the programs executed for performing an automated step. Many of these scripts execute *transactions* against a backend database. Thus, humans communicate with the workflow engine through a browser and the workflow engine communicates with the web servers directly using the HTTP protocol. Essentially the workflow engine becomes a client of the Web servers. Figure 2 portrays the equivalent components of a WFMS on the Web.

4 Ingredients of Efficient Solutions

Executing workflows over the web necessitates development of solutions to the following problems [KR97]:

1. Unreliability of the Web — network failure/delay, server overloading
2. Heterogeneity of data available from different Web servers
3. Ensuring workflow progress and trackability in spite of distributed execution of workflows.

Because the Web has become a standard platform for disseminating information any new mainstream application has to *adhere to the standard infrastructure*. This implies that the solution proposed to these problems should conform to the HTTP protocol, the HTML language, and should not require code changes for Web servers or client browsers. Whereas a centralized scheduler simplifies tracking and ensuring progress of workflows multiple such schedulers can be used in parallel for load balancing and for achieving scalability. Consider a large sequence of steps, each to be performed at a different external Web server. After the completion of each step, the status is recorded in the workflow database and this simplifies tracking. To deal with data heterogeneity, we need a facility for plug-in syntactic translators [KRGL97]. Also, we need protocols to ensure that an update request is processed by a server exactly once, despite network/server failures [KRGL97]. This will facilitate the reliable execution of workflows on the Web.

The design gets simplified if we assume that Web server assigned for each automatic step will provide proper interfaces for performing the various functions (*e.g.*, placing a request, enquiring about the status of a request, canceling a request, modifying a request). The Web interfaces access data from backend DBMSs using a CGI script or some other method (*e.g.*, the direct interface between HTML and SQL described in [NS96]). Each step in a workflow accesses data from at most one Web server.

5 Applications of World Wide Workflows

In this section, we first show how workflows can be used to build virtual warehouses. Subsequently, we show how with such a warehouse in hand, electronic commerce applications can be built.

5.1 Creating Virtual Warehouses

We have designed and implemented an application called *StockAdvisor* to illustrate the concept of using workflows for data collection in data warehousing environments. The application is a tool for making "well-informed" stock investment decisions.

With StockAdvisor, once a user selects a company of interest he/she is provided with (i) corporate information about the company (ii) stock performance data of the company and its major competitors over the past year and (iii) stock performance over the past year of all the different sectors of the economy. StockAdvisor connects to different

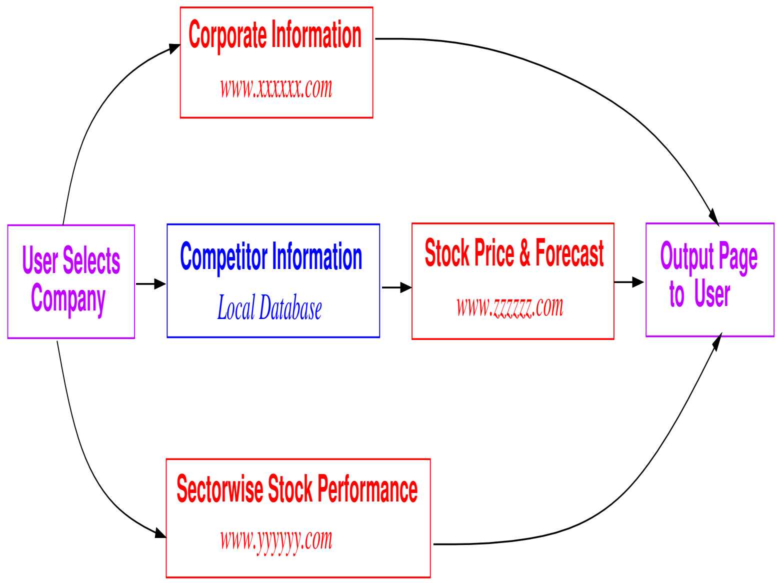

Figure 3: Virtual Warehousing – StockAdvisor Schema

local databases and remote web servers (each being a step of a workflow) that provide information on different specialties. The information received is then filtered, processed, and then consolidated and presented to the user in an easy to read manner. The information presented to the user consists of text, tables, graphs and bar charts. The bar charts are applets created on the fly based on the processed information, while the graphs are directly obtained from the source.

Thus, StockAdvisor helps automate the process of bringing the necessary data to provide the specific information needed by a user. Specifically, when a user clicks on a company link, at the web server a `cgi-bin` script is invoked that triggers a workflow. The *virtual warehouse* that results consists of information that is periodically updated using data from different web information sources (push method of data collection). Since the workflows themselves are executed in a reliable manner, if a web server is not reachable, data can be fetched from an alternative source or by repeatedly retrying with the original server.

Currently, the stock price information is static, i.e., it only shows the price of the stocks when the page was generated. What would be more interesting and useful is data refreshing schemes by which stock prices can be shown dynamically to the user as they change. Research on incremental updates of data warehouses will be useful in developing the data refreshing schemes.

StockAdvisor illustrates how work on integrating workflows and warehousing will improve the ability of a user to specify the steps involved in information gathering – with the system systematically performing the steps, thereby automating the process of bringing the necessary data to meet the specific demands of a user. This will increase the accuracy and timeliness of the results made available to users.

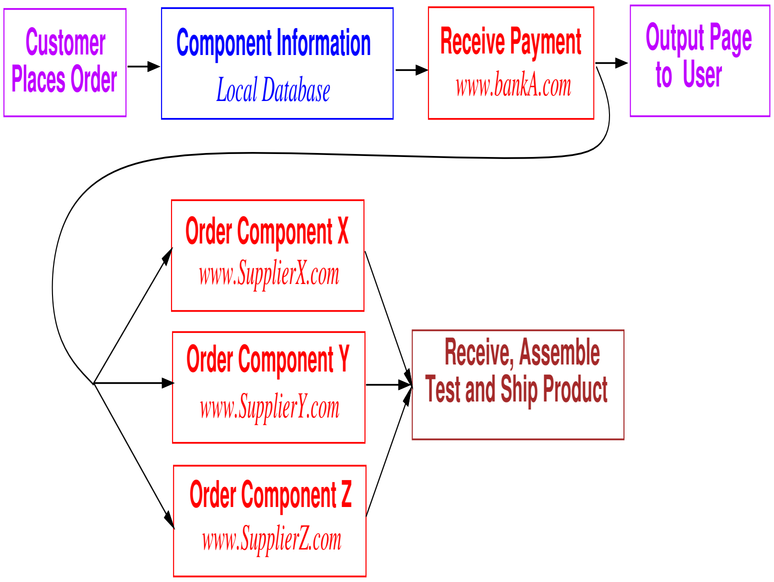

Figure 4: Virtual Enterprise – Order Processing Schema

5.2 Creating a Virtual Enterprise

Suppose a manufacturer builds systems from components obtained from outside sources. The manufacturer first collects all the needed information, using a system such as the StockAdvisor. Using this information he determines which sources provide the needed components, places orders for the components after payment is received, collects the components, manufactures the system and ships the product. The schema for the workflow that accomplishes these steps is portrayed in Figure 4. The step "Component Information" corresponds to a workflow by itself to build the needed virtual warehouse with information about providers of components.

Such a system enables business-to-business transactions across enterprises and permits round the clock order processing. Furthermore, using the facilities provided in [KRGL97], we can achieve reliable (handling failures and contingencies) and responsive (reduced latency) processing of the orders.

6 Conclusion and Related Work

Although commercial efforts for integrating web technology with workflow technology have mainly focused on using the Web as an interface to WFMSs, to take full advantage of the web's possibilities, it is necessary to create WFMSs that can understand the Web and utilize the information available on the Web to automate business processes. To this end, we discussed technologies that can be used to build workflows that can directly query/update the online information available on the web.

Before concluding the paper, we discuss related work.

The Meteor$_2$ project [S$^+$96] has implemented workflows on the Web. In this approach, a step typically consists of a form that is filled in and processed by a CGI script which in turn generates a form with the URL of the CGI script that should perform the next step of the workflow. This process is repeated to complete the workflow. Thus, there is no central workflow engine/scheduler to control and manage the workflow. The workflow schema is essentially embedded in the different CGI scripts which are responsible for performing the control flow. Data is passed between steps through the hidden fields of the forms. Optionally CGI scripts communicate with Web servers using sockets and the HTTP protocol to execute other CGI scripts for automatically accessing databases. Problems of reliability and efficiency exist even in this architecture. Even though they do not use a central scheduler, when accessing Web servers external to the organization, the gateway can be a bottleneck when there are large number of workflows. In the event of a failure/exception, it is possible for some workflows to stall. Several sites may have to be traversed before the cause and the location of a stalled workflow step is determined. Since workflows cross organization borders, it is difficult to ensure progress and trackability.

Our work also differs from the numerous recent research efforts to query information from multiple data sources on the web [ACHK94, LRO96]. Their aim is to determine which sites to get the information from and how to process the obtained information to satisfy a given query. If some of the sources in the list of data sources are not accessible, it is not much of a concern since the user will be happy with whatever data is returned and will not be able to differentiate the case where some data sources are inaccessible. In contrast, a workflow has a definite sequence of steps to be executed at different servers to complete a workflow. Executing them reliably over the unreliable Web is an important requirement. However, some of the data modeling techniques developed for querying can be enhanced for automatically constructing workflow schemas. For example, sophisticated data modeling and search mechanisms like the ones developed for the Information Manifold [LRO96] might be used to construct a workflow schema that consists only of automatic steps and performs data mapping from the output of one step to the input of another using the built-in knowledge (ontology).

Electronic brokerages where numerous value added services are offered by a single agency with the ability to track and account for each order placed by a customer are becoming popular. Much effort is being dedicated to the development of Electronic Data Interchange (EDI) standards and Business APIs (*e.g.*, SAP R/3 developed by an alliance of SAP and Microsoft) to facilitate electronic commerce. These trends indicate the importance of our work in developing tools to build global cooperative information systems for conducting business over the web. Our work is also relevant in the context of performing distributed computation over the Web. The objective of both the systems is to visit specific Web sites to access and update information to achieve a defined goal. Thus, an important by-product of our work is a way to reliably program the Web with its current infrastructure, thereby realizing the goal of building a true global information system.

References

[Act] *Action Workflow Metro World-Wide-Workflow*, Action Technologies, Inc. `http://www.actiontech.com/metrotour/resources/Metwp.htm`.

[ACHK94] Y. Arens, C. Chee, C. Hsu, and C. Knoblock. Retrieving and Integrating Data from Multiple Information Sources. *Journal of Intelligent and Cooperative Information Systems*, 1994.

[Flo] *IBM FlowMark*, `http://www.torolab.ibm.com/workgroup/flowmark`.

[Geo95] Georgakopolous D. and Hornick M. and Sheth A. An Overview of Workflow Management: From Process Modelling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, 3(2):119–152, April 1995.

[Hol94] D. Hollingsworth. The Workflow Reference Model. The Workflow Management Coalition, http://www.aiai.ed.ac.uk/WfMC/, 1994.

[KR97] M. U. Kamath and K. Ramamritham, Correctness Issues in Workflow Management, *Distributed Systems Engineering Journal*, special issue on Workflow Management, Volume 3, Number 4, December 1996, pp. 213-221.

[KRGL97] M. Kamath, K. Ramamritham, N. Gehani, and D. Lieuwen. WorldFlow: A System for Building Global Transactional Workflows. In *(to appear in) Proc. of 7th Intl. Workshop on High Performance Transaction Systems (HPTS)*, 1997.

[HTT] *Standard for Hypertext Transfer Protocol – HTTP/1.0 and HTTP/1.1*, http://www.ics.uci.edu/pub/ietf/http/.

[Java] *The Java Platform*, http://www.javasoft.com/aboutJava/index.html.

[Javb] *Netscape JavaScript Authoring Guide*, http://home.netscape.com/eng/mozilla/Gold/handbook/javascript/index.html.

[LRO96] A. Levy, A. Rajaraman, and J. Ordille. Querying Heterogeneous Information Sources Using Source Descriptions. In *Proceedings of the 22nd VLDB Conference*, Bombay, India, 1996.

[NS96] T. Nguyen and V. Srinivasan. Accessing Relational Databases from the World Wide Web. In *Proc. of ACM SIGMOD*, pages 529–540, Montreal, Canada, 1996.

[S+96] A. Sheth et al. Supporting State-Wide immunization Tracking using Multi-Paradigm Workflow Technology. In *Proceedings of the 22nd VLDB Conference*, Bombay, India, 1996.

[She96] A. Sheth, editor. *Proceedings of NSF Workshop on Workflow and Process Automation in Information Systems: State-of-the-Art and Future Directions*, http://optimus.cs.uga.edu:5080/activities/NSF-workflow/proc_cover.html, May 1996.