The Future of Web-Based Workflows

John A. Miller, Amit P. Sheth, Krys J. Kochut and Devanand Palaniswami
Large Scale Distributed Information Systems Lab (LSDIS)
Department of Computer Science
The University of Georgia
Athens, GA 30602-7404
email: <*jam,devanand,amit,kochut*>*@cs.uga.edu*
URL: *http://LSDIS.cs.uga.edu*

ABSTRACT

In this position paper, we discuss the suitability of the Web as the primary infrastructure for supporting advanced workflows in heterogeneous, distributed environments in which security, reliability, high performance and transactional capabilities are important. At present, CORBA provides many advantages to Web technology as well as some disadvantages. However, as Web technology evolves to include CORBA-like capabilities some of these advantages will likely diminish in the future. In addition, with capabilities like IIOP and Java to CORBA interfaces the two technologies can be combined where needed. The METEOR project at the University of Georgia is exploring all of these angles with WebWork, a purely Web-based Workflow Management System (WfMS), with OrbWork, a primarily CORBA-based WfMS with Web interfaces, and by the the fact that WebWork and OrbWork will conveniently interoperate. In this paper, we will highlight several issues in which current Web technology is lacking, but future developments hold promise. In some cases weaknesses in the infrastructure can be compensated by implementing capabilities at a higher level in the WfMS. Generally, however, these approaches are less desirable than having the capability at the infrastructure level.

1 Introduction

Workflow Management Systems (WfMSs) provide an automated framework for managing intra- and inter-enterprise business processes. According to the Workflow Management Coalition (WfMC), a Workflow Management System is a set of tools providing support for process definition, workflow enactment, and administration and monitoring of workflow processes [Hol94]. Application domains where workflow technology is currently in use includes healthcare, education, telecommunications, manufacturing, finance and banking and office automation. WfMSs are being used today to re-engineer, streamline, automate and track organizational processes involving humans and automated information systems [JAD+94, GHS95, Fis95, SKM+96, SGJ+96]. The success of WfMSs has been driven by the need for businesses to stay technologically ahead of the ever-increasing competition in typically global markets.

WfMSs are very complex pieces of software. Supporting advanced capabilities such as security, reliability, high performance and transactional capabilities in heterogeneous and distributed environments in a manner that facilitates interoperability with other WfMS as well as other software systems and tools almost demands use of an appropriate and widely-used infrastructure. In the METEOR_2 project, two principal infrastructures - Web (World Wide Web) and CORBA (Common Object Request Broker Architecture) - are used to build WfMSs, WebWork and OrbWork, respectively. These infrastructures were carefully chosen for their functionality, availability, popularity and reasonable cost structures. Web technology is appropriate in two distinct and important ways.

First, the ubiquitous nature of Web browsers makes them a natural user interface. Web browsers satisfy one of the primary concerns in application deployment - it allows users with any of the popular computing platforms to be able to participate in a workflow without any additional hardware. A multitude of users (many of whom are not computer sophisticated) are already familiar with these browsers' easy-to-use interface. They presently see the interface as a way to access all sorts of information and perform simple tasks such as filling out forms. The unformity, wide availability and simplicity of the interface makes Web browsers an ideal user interface for workflow applications. This is particularly true in the healthcare domain where there is little time or inclination for special-purpose training.

Second, Web technology provides a solid communications infrastructure for building WfMSs. For WebWork, it is the only communications infrastructure, while for OrbWork it plays a supporting role to CORBA. We view CORBA as a key element for building very robust transactional workflow systems [SKM+96]. However, we believe it is unlikely that all organizations participating in a workflow (clinics and small hospitals, for example, in the healthcare domain) will want to purchase a CORBA product or, more importantly, will wish to maintain the software. It is more likely that they will have a Web server or at least access to a Web server. For this reason, METEOR$_2$ can support workflows in a Web-only (CORBA-free) mode. CGI programs/scripts are used to run tasks and coordinate the overall execution of workflows. The WebWork implementation relies on Web browsers, Web servers, HTML, JavaScript and CGI.

Many workflow management systems provide Web interfaces, but use other mechanisms for underlying communication/distribution (e.g., sockets, rpc or CORBA). Typically, these involve existing non-Web based engines for which Web interfaces are layered on top. In many cases, the workflow engine is centralized, with only access being distributed (multiple clients, single server). In the case of multiple servers, daemon processes must be installed and kept running at multiple sites for the system to work. We call workflow management systems that provide Web interfaces *Web-enabled*, while if Web technology is the only infrastructure used to build the workflow management system, providing both interfaces and communication/distribution, we say that it is *Web-based*.

A survey of existing commercial workflow systems that use Web technology [CS897], [SJ96], [PLS+96], [AAAM97] revealed that a majority of them are merely Web-enabled. But the trend appears to be to add more and more Web orientation to the WfMSs. This trend is illustrated by the following systems: ActionTech Metro, WebFlo, DartFlow, OzWeb, Endeavors and WebWork. Each one is successively more Web oriented, with WebWork being purely Web-based. A couple of other notable systems that utilize the Web and provide workflow capabilities are WebFlow [Cor97] and PrISMS [NAS96].

- **ActionTech Metro** [Tec97] by Action Technologies is a web-enabled workflow product, which supports many of the regular office automation processes. This includes passing of messages and automatic e-mails, keeping track of actions carried out by various users, and supporting the role paradigm. The Metro product is a Web interface to Action Technologies' conventional ActionWorkflow family of workflow products. Organizations such as Sandia National Laboratories are choosing products like ActionTech Metro, because its Web interfaces greatly reduce deployment costs.

- **WebFlo** [Con97] by Information Management Consultants consists of a useful product suite supporting Web interfaces to WfMSs (currently FileNet) as well as allowing the insertion of Java or CGI applications into work item forms to perform special functions. The product suite includes the following modules: WebFlo.Initiate creates work items to be processed through workflow. WebFlo.Status queries queues or tables to determine the status of work-in-process. WebFlo.Work allows users to participate in an existing workflow over the Internet. WebFlo.Docs provides document retrieval applications to view images stored in an organization's imaging system. WebFlo.AppBuilder is a windows-based wizard tool for creating custom applications. WebFlo.Secure is a windows-based tool for administering user privileges.

- **DartFlow** [CGN97] is a research prototype being developed at Dartmouth College. Their goal is to utilize Web technology to a greater extent than current WfMSs. They nicely summarize this in their paper [CGN97].

  "Unfortunately, WWW integration is mostly limited to offering a web browser interface to different proprietary workflow engines. In this paper we propose a different approach using open and portable Web technology not only as a front end for the workflow client applications, but also for the implementation of the workflow enactment service and for administration and monitoring."

  It uses Web-browser embedded Java applets as its front end and transportable agents as the back end. Java provides platform independence, and transportable agents are used to make DartFlow flexible and scalable. Once a user's password is verified the user is logged in and his/her worklist comes up. By clicking on the worklist, an HTML form is displayed. When it is submitted, a CGI script is invoked that creates a new agent to process the form. Since each transportable agent contains its own process description, DartFlow has the capability to adapt each instance of a task to its specific needs. The agent lives until it finishes the task for which it was created. DartFlow relies on Agent Tcl, also being developed at Dartmouth, in addition to Web technology to provide its foundation. It requires Agent Tcl servers to be running at participating sites and communication is provided by Agent Tcl using for example message passing or remote procedure calls (RPC).

- **OzWeb** [KDY97] supports software engineering processes and workflow and is an extension to Oz that operates on Web entities. Since OzWeb is Web-based it provides a significant advantage over its predecessors, Oz and Marvel, in that it can be easily adopted by an organization as it requires no changes to either Web servers or browsers.

- **Endeavors** [Tay97] is a process/workflow system that is Web-based and is evolving toward providing all of the following characteristics: (1) it is built using Java and fully utilizes Web protocols and tools; (2) it is model-based; (3) it supports dynamic workflows; (4) it is integrated at a fundamental level with hypermedia; and (5) it attempts to make the "process" as invisible as possible.

- **WebWork** is a purely Web-based implementation of the METEOR_2 Workflow Model. We consider the fact that WebWork has been implemented as a functionally complete WfMS exclusively using Web technology as its communication infrastructure to be one of its major advantages. Test runs using WebWork have confirmed our original assumptions in choosing pure Web technology as the basis for WebWork. First, the WebWork WfMS was easy to install at all our test sites, with minimal changes to system configuration files. To begin developing workflow applications using WebWork, all that was required was for the associated Web server to be up and running. Second, the workflow application development process was very straightforward. After using the METEOR_2 graphical workflow designer to design the workflow application, the application developer needs to use the WebWork application builder which consists of a couple of simple Web forms. This step typically requires very little user interaction to finalize/customize the specifications. These specifications are used to generate the run-time code for the application. Third, installing the application even across a distributed computing environment was straightforward and required human intervention only in terms of transferring tarred files and running installation scripts. WebWork supports the development of workflow applications that can run in heterogeneous and distributed environments. Any number of organizations can be easily incorporated into a workflow - a Web server at an organization accesses its local database(s) using CGI programs, and these CGI programs can be accessed using Web browsers at various other organizations - thus facilitating a distributed client-server implementation. In addition, multiple Web servers are needed to integrate the various autonomous organizations and databases in distributed workflow applications. CGI programs can interact with existing heterogeneous DBMSs. Thus, already existing infrastructure can be efficiently utilized. Moreover, organizations participating in a workflow can easily use existing hardware, as Web-based systems are highly platform independent.

2 Issues

At this point, we would like to present some important issues in the future development of Web technology that are crucial to its role as an infrastructure for WfMSs.

- **Location Transparency**. At present WebWork supports location transparency by having each task manager examine a file called cgi_location.map which specifies where (e.g., the URL) other task managers are located. CORBA provides this capability since object names and locations are registered with the ORB. In the near future, this capability will be provided by the Web.

- **Object-Orientation**. OrbWork allows arbitrary CORBA data objects to be sent between tasks. The objects may have attributes, methods and relationships with other objects. WebWork does not support arbitrary data objects, but rather sends data tuples (analogous to the relational data model) between tasks. A data tuple consists a number of attribute values where each attribute values must come from an atomic (non-composite, and single-valued) domain. If, for instance, an image needs to be sent, its URL (a simple string) is included in the data tuple.

- **Error Handling and Recovery**. A workflow represents a very complex computational activity. There are many tasks whose execution needs to be coordinated. Some of these tasks may be newly developed and unfortunately not tested as thoroughly as they should be. Also, resources may be temporarily unavailable and the workflows must adapt to this. Therefore, comprehensive error handling and recover mechanisms should be included with any industrial strength WfMS. Any WfMS must handle common errors and failures of tasks if it is to be of practical use. In addition, error handling and recovery at the task manager and workflow levels are also important [WS97]. WebWork's goal is to provide useful and practical error handling and recovery with minimal overhead. Presently, the error handling and recovery mechanisms simply rely on three types of persistent data: (1) data maintained by the Web servers, (2) data in the WebWork worklists or input logs (one for each task), and (3) data recorded in WW_error_logs (one for each Web server). For WebWork, we have classified ten types of errors or failures that may be encountered ranging from data entry errors on forms to web server failures [MPS+97].

- **Transactional Capabilities**. Although individual tasks such as database updates may be transactional, a sequence or set of database updates on multiple databases, for instance, is not supported as transactional. This requires advanced transactional support (e.g., nested transactions, two-phase commit, sagas, etc.) from the WfMS. Providing such capabilities in a general-purpose fashion are quite difficult, requiring at least the capabilities of a Transaction Processing Monitor (Encina or Tuxedo). CORBA implementations are just now beginning to provide full implementations of the Object Transaction Service (OTS). Our OrbWork implementation is making use of Iona's Orbix OTS based on Encina. Transactional capabilities will be showing up both in Web servers and in Java in the near future. Web servers providing atomic transactions will be quite useful for WebWork. For example, consider a transactional task which updates a database and then prepares a page to output the results. It is possible that the database may be updated, but a failure occurs before the page can be output. Consequently, the task viewed as whole is not atomic, since it accomplished only one of its two major parts. In the longer term, the Web Distributed Authoring and Versioning (WebDAV) project [GEWF+97] will provide other useful capabilities, such as locking and versioning of Web resources.

- **Security**. Since the Web is becoming a medium for electronic commerce, sophisticated mechanisms for security, authentication and access control have and are being developed (e.g., SSL, HTTP-S, Digital Signatures and WebDAV). These will need to be used as a foundation for security in Web-based WfMSs.

- **Performance**. Performance can be enhanced in a variety of ways including the use of a cluster of Web servers with load balancing, better techniques for running tasks in parallel, newer http protocols (e.g., the recently introduced HTTP/1.1), WebNFS, etc. In addition, use of Java or JavaScript allows for a type of load balancing between clients and servers.

3 Summary

The philosophy behind the development of METEOR_2 implementations is that they should be built on top of powerful, yet commonly used infrastructures. The Web was chosen because of its ubiquitous, low cost and easy-to-use nature. CORBA was chosen because of its powerful distributed object management (e.g., it is able to support nested transactions in distributed and heterogeneous environments). One of the highlights of the METEOR_2 approach is that advanced transactional capabilities will be provided by the implementations. This requires at least the capabilities of a Transaction Processing Monitor (TPM) to be provided by the WfMS. Because of the complexity of this we are relying on the infrastructure to provide much of these capabilities (e.g., CORBA Object Transaction Services). The intent of WebWork is to provide most of the capabilities of its CORBA counterparts in a smaller and lower cost package. As Web technology evolves, we believe that the case for using it as a primary infrastructure for building WfMSs becomes stronger. In particular, as the issues discussed above are addressed, its future will be quite promising.

Acknowledgements. This research was partially done under a cooperative agreement between the National Institute of Standards and Technology Advanced Technology Program (under the HIIT contract, number 70NANB5H1011) and the Healthcare Open System and Trials, Inc. consortium. See URL: *http://www.scra.org/hiit.html.* Additional support and donations are provided by Visigenic, Informix, Iona, Hewlett-Packard Labs, and Boeing.

References

[AAAM97] G. Alonso, D. Agrawal, A. El Abbadi, and C. Mohan. Functionalities and Limitations of Current Workflow Management Systems. Technical report, IBM Almaden Research Center, 1997. To appear in IEEE Expert.

[CGN97] T. Cai, P. Gloor, and S. Nog. DartFlow: A Workflow Management System on the Web using Transportable Agents. Technical report, Dartmouth College, 1997. URL: http://www.cs.dartmouth.edu/reports/Cai,Ting.html.

[Con97] Information Management Consultants. WebFlo - Delivery Imaging and Workflow over the Web. Technical report, Information Management Consultants, 1997. URL: http://www.imcinc.com/webflo.

[Cor97] WebFlow Corporation. Putting your team on the SamePage. Technical report, WebFlow Corporation, 1997. URL: http://www.webflow.com/product.html.

[CS897] CS835. Enterprise Integration Course Technical Reports. Technical report, University of Georgia, Department of Computer Science, 1997. URL: http://lsdis.cs.uga.edu/ amit/EI.html.

[Fis95] L. Fischer. *The Workflow Paradigm - The Impact of Information Technology on Business Process Reengineering*. Future Strategies, Inc., Alameda, CA, 2nd. edition, 1995.

[GEWF+97] Y. Goland, Jr. E.J. Whitehead, Asad Faizi, S.R. Carter, and Del Jensen. Extension for Distributed Authoring and Versioning on the World Wide Web – WEBDAV. Technical report, UCI, March 1997. URL: http://www.ics.uci.edu/ ejw/authoring/draft-jensen-webdav-ext-01.txt.

[GHS95] D. Georgakopoulos, M. Hornick, and A. Sheth. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, 3(2):119–154, April 1995.

[Hol94] D. Hollingsworth. The Workflow Reference Model. Technical Report TC00-1003, Issue 1.1, The Workflow Management Coalition, Brussels, Belgium, November 1994.

[JAD+94] S. Joosten, G. Aussems, M. Duitshof, R. Huffmeijer, and E. Mulder. *WA-12: An Empirical Study about the Practice of Workflow Management*. University of Twente, Enschede, The Netherlands, July 1994. Research Monograph.

[JK97] S. Jajodia and L. Kerschberg, editors. *Advanced Transaction Models and Architectures*. Kluwer Academic Publishers, 1997. To be published.

[KDY97] G.E. Kaiser, S.E. Dossick, and J.J. Yang. Process Enactment for the World Wide Web. Technical report, Columbia, 1997. URL: http://www.psl.cs.columbia.edu/ozweb.html.

[MPS+97] J. A. Miller, D. Palaniswami, A. P. Sheth, K. J. Kochut, and H. Singh. WebWork: METEOR_2's Web-based Workflow Management System. Technical report, University of Georgia, April 1997.

[NAS96] NASA. Program Information Systems Mission Services (PrISMS). Technical report, NASA, 1996. URL: http://ec.msfc.nasa.gov/prisms.

[PLS+96] D. Palaniswami, J. Lynch, I. Shevchenko, A. Mattie, and L. Reed-Fourquet. Web-based Multi-Paradigm Workflow Automation for Efficient Healthcare Delivery. In *[She96]*, Athens, GA, May 1996.

[SGJ+96] A. Sheth, D. Georgakopoulos, S. Joosten, M. Rusinkiewicz, W. Scacchi, J. Wileden, and A. Wolf. Report from the NSF Workshop on Workflow and Process Automation in Information Systems. Technical report, University of Georgia, UGA-CS-TR-96-003, July 1996. URL: http:// LSDIS.cs.uga.edu/activities/NSF-workflow.

[She96] A. Sheth. Proc. of the NSF workshop on workflow and process automation in information systems. University of Georgia, May 1996. URL: http://LSDIS.cs.uga.edu/activities/NSF-workflow.

[SJ96] A. Sheth and S. Joosten. Workshop on Workflow Management: Research, Technology, Products, Applications and Experiences, August 1996.

[SKM+96] A. Sheth, K. J. Kochut, J. Miller, D. Worah, S. Das, C. Lin, D. Palaniswami, J. Lynch, and I. Shevchenko. Supporting State-Wide Immunization Tracking using Multi-Paradigm Workflow Technology. In *Proc. of the 22nd. Intnl. Conference on Very Large Data Bases*, Bombay, India, September 1996.

[Tay97] R.N. Taylor. Dynamic, Invisible, and on the Web. Technical report, UCI, 1997. URL: http://www.ics.uci.edu/pub/endeavors.

[Tec97] Action Technologies. Metro 3.0 Coordinates Work across the Web. Technical report, Action Technologies, 1997. URL: http://www.actiontech.com/Metro.

[WS97] D. Worah and A. Sheth. Transactions in Transactional Workflows. In *[JK97]*, chapter 1. Kluwer Academic Publishers, 1997.