Endeavors and Component Reuse in Web-Driven Process Workflow

Arthur S. Hitomi
Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425 USA
+1 (949) 824-4101
ahitomi@ics.uci.edu

Dong Le
Kofax Image Products
3 Jenner Street
Irvine, CA 92618-3802 USA
+1 (949) 727-1733 ext 283
dle@kofax.com

ABSTRACT
Processes that perform and distribute tasks over the Internet are known as "Web-Driven Process Workflow." The *UCI Travel Expense Process (TEP)* and the *Pacific Bell Applications Development Group (ADG)* are two web-driven process workflows that exemplify process reuse by leveraging Endeavors, a research workflow process support system at the University of California at Irvine. This paper describes the TEP and ADG projects, and the enabling components and architecture that allow reuse and extension of process components with off-the-shelf tools and web-based resources over the Internet.

Keywords
Workflow, process, Internet, components, reuse, architecture

INTRODUCTION

The UCI Travel Expense Process (TEP) and the Pacific Bell Applications Development Group (ADG) [3] are two examples of web-driven process workflows. Both systems involve common workflow activities such as review and approval processes, and both provide distributed process and information across groups and individuals. To build reusable web-driven process workflows required two major components 1) adding web extension to the architecture of an existing turnkey workflow process system and 2) developing necessary components and tools for creating and incrementally adopting existing web infrastructure into a process workflow environment.

The Endeavors system [1] was chosen as the process workflow environment. It provides an object oriented component model, allowing for reuse of components and process objects such as activities, artifacts, and resources associated with a software development process. It also consists of a layered virtual machine architecture with three major levels. Of the three, the middle and top levels (System and User) were augmented with a specialized HTTP server and POP3 email client through a major component, WebNavigator. The WebNavigator was designed and developed to provide a bridge between Endeavors and the Web server, relaying information and interpreting data structures between the two systems.

Once the system was developed and integrated, we designed the process using the Endeavors visual process notation and object model. Since the ADG and the TEP had many similar activities and process goals, we leveraged Endeavor's reuse mechanisms and WebNavigator's extensions for sharing components over the World Wide Web.

TEP AND ADG AS DISTRIBUTED WORKFLOW

Pacific Bell Network Operations has no "process" for developing technical applications and for this reason the ADG or Applications Development Group was created. The ADG is a system that automates the phases of software project design over multiple platforms over a wide area network. The system lets clients access workflow applications and formalizes steps, leading from a basic concept to a formal proposal for development. Overall, the ADG provides enough structure to allow clients, developers, and administrators to create, monitor and develop software projects.

The application development process must satisfy three basic client requirements:

- It must make the application developers accessible to the clients in terms of discussing their needs and developing their ideas.
- It must formalize the steps leading from a basic "concept" to a formal proposal for development.
- It must provide enough structure to establish accounting and tracking mechanisms for all process and application artifacts.

The Travel Expense Process (TEP) is a document routing and approval process for reimbursing faculty, staff and students on University funded trips. Unfortunately, the current solution to this process generates large amounts of paperwork and traffic within the organization. The goals of the TEP is to automate the review and approval process in order to 1) save significant time, and 2) coordinate efforts required by the approval committee to process travel

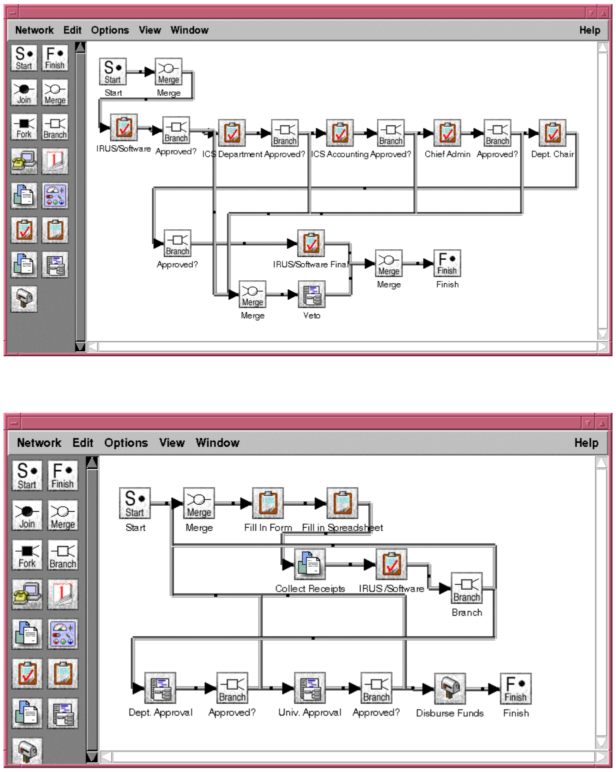

Figure 1. Top - The TEP Activity Main Networks. Bottom - The Sub-Network of the Department Approval

expense documents.

The routing, tracking, and managing of travel documents by an organization of UCI's size is not unique. Both large and small organizations have this problem disguised in other ways. For example, a purchase order (PO) request at companies usually involves a similar set of forms (the purchase order form) with similar activities (purchase order request and purchase order approval). Typical solutions to this problem often fail for two reasons: 1) They are usually built on top of databases, an effective but very timely and inflexible solution that requires a high number of stakeholders to justify the high cost of development and maintenance and 2) most fail to provide effective process tools for all stakeholders participating in a system's evolution. The ideal process system requires user involvement in evolution-capable processes, visibility into product and process state, use of tools and interfaces that are effective for both technical and non-technical users, and mechanisms for facilitating customization and dynamic change.

The TEP and the ADG share several common process goals. Both process workflows require easier tracking of document artifacts over a (long) period of time and require electronic routing of artifacts from stakeholder to stakeholder. To automate this task would greatly improve the efficiency and ability for users to access the systems over a distributed network or over the World Wide Web. Though a sophisticated workflow management system may satisfy some of these requirements, important issues of incremental adoption and low entry barrier are unresolved when working with current workflow process and database technologies[1].

ARCHITECTURE OVERVIEW

Figure 1 shows a high level view of the system architecture and functional breakdown. Several key components and systems are highlighted in this figure: The Java Web Server, Web Navigator, and Endeavors.

Endeavors employs several key features of an ideal workflow process system. Two of the most important are the ObjV object oriented component model [13] and layered virtual machine architecture. ObjV allows for reuse and evolution of components and process objects. The other, the layered virtual machine allows for extension of process and user components by layering the architecture with three major levels: The user, system, and foundation.

The user level is responsible for maintaining consistent views to users through management of coordinated updates. It tracks and displays system level changes using event registration and broadcast. The user interfaces provide a visual means for stakeholders to customize the system's objects and configure the system's networks, and offers a set of individual software components known as activity objects for defining activities. Although these system or process objects are defined at design time (before process runtime), their definitions and behaviors can be dynamically specified and modified at runtime. The behaviors are implemented through handlers which are programmable in different languages and environments. The visual network editor shows a control flow perspective of a process. Arcs, control structures and logical paths between activity objects define the perspective. This view allows all stakeholders to participate in the process: System programmers, process programmers, and end users can view and invoke object events, construct and modify processes, and diagnose interpreter execution from a single interface.

The system level maintains the category object model abstractions and data structures. Components of the system level 1) interpret activities and activity networks, and 2) provide mechanisms to distribute, manage, and coordinate workflow. Object storing and handler invocation occur at the foundation level where information is stored and retrieved to and from the object-database which can either be configured to the machine's native file system or a scalable SQL database.

The Java Web Server [9] provides HTTP services across the Internet. This server, like Endeavors is cross platform through the use of Java technology [3]. The Java Web Server uses an application programming interface to provide its HTTP services to external Java components known as *servlets*.

The WebNavigator is the key component for the Endeavors and the HTTP server integration. WebNavigator coordinates multiple interpreters and processes, and with WebInterpreter extends the Endeavors object handlers by integrating HTTP services at the User level. The augmentation of an external web server to Endeavors scaled very well despite the heavyweight nature of the Java Web Server. Listed below are some of the basic functions of the WebNavigator:

- Initiates Endeavors for starting the process.
- Coordinates multiple interpreters by translating HTTP request between the Java Web Server and the WebInterpreter.
- Conveys process information between end users through an extended web-enabled workflow process handlers.
- Delegates and automates activities from one stakeholder to another.

The WebInterpreter is the complement to WebNavigator. It is built on top of the Endeavors interpreter and at execution time traverses the network as well as generates and delivers events to the appropriate object handlers. These events to handlers provide general and web-specific information about requests, delivery and resources. In essence, WebInterpreter is an extension of the standard Endeavors System Interpreter for web environments.

Event Notifications

Workflow needs both push and pull mechanisms to send and receive workflow content to stakeholders. In the WebNavigator system, web browsers receive or pull content from the process onto the stakeholder's desktop over HTTP. They pull web pages by initiating requests using hyperlinks to start or continue an ongoing process. The hyperlink traversal results in initiating WebNavigator, sending the message to the appropriate handler, and then returning the result back to the end user. The hyperlinks can be stored or transferred to another resource, giving end users the convenient option to delay or reroute workflow activities.

While web browsers are used to pull information to the desktop, email pushes out to desktops to notify stakeholders of their awaiting tasks. We examined several technologies for providing push, including the new specification in HTTP/1.1 [2], commercial services such as Castanet [6] and Pointcast [8], and Endeavors' own services through its C2 architecture [11]. Though any of these technologies are easily adaptable, standard email is a familiar tool that end users can use to receive non-intrusive notification. Like the web browser, email is ubiquitous Internet technology available to many platforms and devices and also provides low entry barrier and transparency, two important attributes for adopting workflow systems into environments [10].

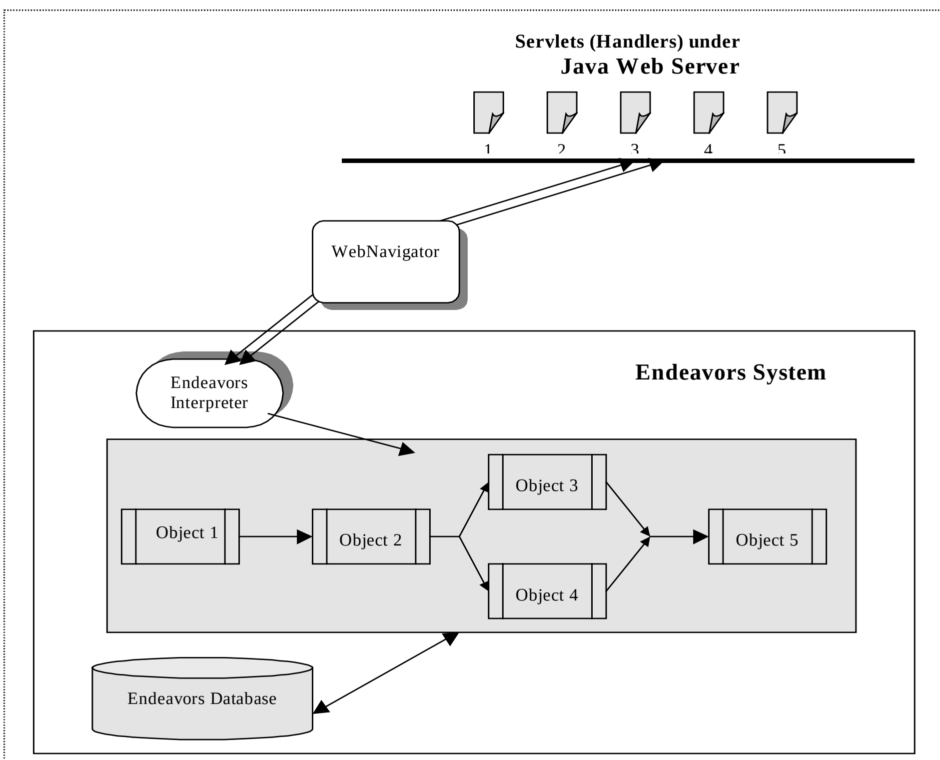

Figure 1. The system architechture overview

External tool integration

Not all workflow tools and resources are accessible nor appropriate for use over the Internet. For example, legacy financial databases in corporations may not be suitable for global access for technical or logistical reasons. These resources are integrated through handlers of activity objects. The Knowledge Depot [5], an organizational memory and learning tool (OMOL) was integrated as an external tool in the ADG. Knowledge Depot stores all artifacts during activity execution where they are sorted and then stored into its database.

COMPONENT AND PROCESS REUSE

A workflow process defined in Endeavors can be a single activity within the system or can contain entire networks called sub-networks or sub-activities.

Endeavors allows specialization and reuse of its process objects through its multi-layered category object model. The model is derived from a LISP OOP language called ObjV and supports multiple inheritances for all objects, similar to Smalltalk. However, the Endeavors model differs from traditional OOP languages with dynamic declaration, modification and extension of all field, state variables and object interfaces at runtime. This model also defines roles for each instance level; object categories are created by system programmers (users with technical and programming knowledge), specifications by process programmers (non-technical users who design and manage processes), and instances by end users. Every level has the capability to separate the object's state from its behavior which plays an important role in reusability.

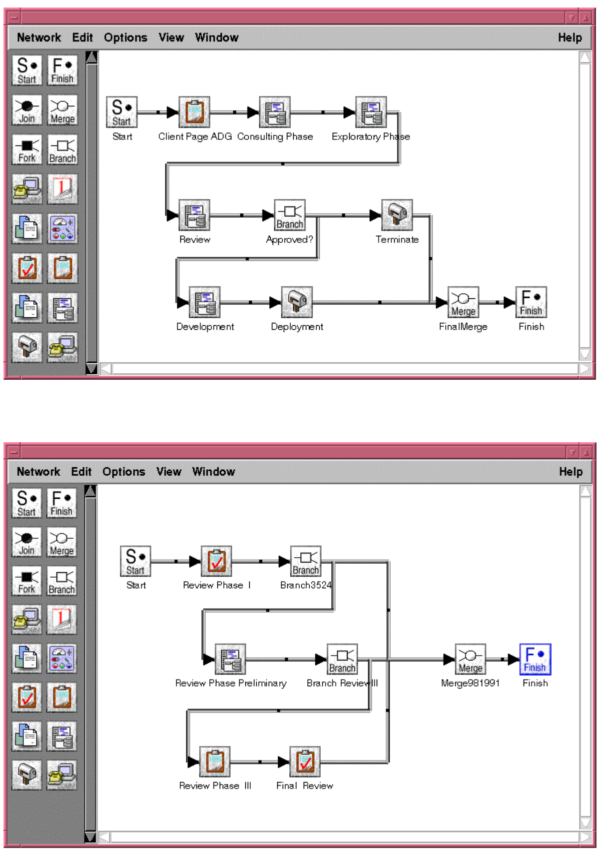

Figure 3. Applications Development Group (ADG). Above is the main activity network. Below is an expanded or subnet view of the Review activity from the main activity network.

Using the Endeavors object framework, we created several reusable tools and components suitable for the ADG and TEP processes.

CATEGORY	SPECIFICATIONS	REUSE
Create Activity	Client Page ADG Create IRS Create IFE Create RFD Create Preliminary Create Final	6
Approve Activity	Review Phase I Final Review Decision Activity Phase II	3
Subnet Activity	Consulting Phase Exploratory Phase Review Development Review Phase Preliminary	5
Disburse Activity	Deployment Terminate	2
Email Hand Off	Email Consulting Email Exploratory Email to Review I Email to Review II Email Review Board	5

Table 1. Components in ADG

The Fill-in-form activity category in TEP and ADG shared several fields and general behavior was specified in a category through object handlers. Once defined, two specifications of the category object were instantiated by a process programmer through the Endeavors network editor. The specifications were generated, creating an instance for each of the projects. For activities to be flexible and reusable, it was important that each activity be very configurable.

Reusing Tool and Resources

Many resources and information can be accessed using the World Wide Web to help aid in workflow systems. Both TEP and the ADG projects included resources such as help desks, instructional handbooks, and online directory assistance. While end users participate in the workflow, they may also refer to other resources or processes outside of their company's Intranet and venture on the web. One of WebNavigator's key capabilities is its ability to incrementally adopt existing web infrastructures into the Endeavors process environment. In the case of the TEP, existing forms, instructions, conversion utilities, and other tools and resources were made available to the stakeholder. ADG and TEP both required special web tools such as the Netscape Enterprise Server [8] and the UCI employee database. Integrating these and other web resources eases access and adoption of existing resources and reduces cost of integration. In addition, reusing web pages lowers entry barrier by allowing incremental adoption of web infrastructure and resources.

For example, every review activity in the ADG project has a parent category with a URL field for a web page address. The review activity object loads a web page approval form from the URL, parses the page to identify or associate “trigger” fields of certain behaviors such as form approval and disapproval buttons, and identifies attached resources contained in the page. After the automation, the end user customizes the web-based activity and behaviors through field parameterizations using a user specification tool for the activity component.

Tables 1, 2 and 3 below lists some examples of objects used throughout the TEP and ADG. Tables 1 and 2 display reuse of category objects (per project) in the left column and the number of its instantiations at design time in the right column. Table 3 displays the numbers of shared components between the TEP and ADG.

Process reuse

Upon process execution, the Endeavors interpreter creates an instance for every object. Instances hold all information pertaining to a single process of execution. Creation or modifications to any object during process runtime are stored within the instance object. This mechanism provides 1) versioning by mapping instance creation over time 2) fine grain control of a single (executing) process by manipulating its instance and 3) execution control of all processes by manipulating specification objects (instance parent). Since all process modifications to networks and objects are stored within instances, instances can be defined to form new process specifications. We introduce *category promotion* for evolving and reusing instances by promoting instances into new process specifications. Re-specifying instances into specifications allow processes to evolve through reuse of collective data.

CATEGORY	NUMBER USED IN ADG AND TEP
Create Activity	6
Approve Activity	10
Subnet Activity	2
Disburse Activity	4

Table 3. Components used in TEP and ADG

CATEGORY	SPECIFICATIONS	REUSE
Create Activity	Fill-In Form Fill-In Spread Sheets	2
Collect Activity	Collect Receipts	1
Approve Activity	IRUS/Software IRUS/Software (in Dept. Approval Sub-network) ICS Department ICS Accounting Chief Admin Dept. Chair IRUS/Software Final	7
Subnet Activity	Dept Approval Univ. Approval Veto Veto (in Dept. Approval subnetwork)	4
Disburse Activity	Disburse Funds	1

Table 2. Components in the TEP

The TEP and ADG shared a similar process. Both consisting of 3 basic steps 1) to create the main artifact and supporting documents, 2) to review and approve all items 3) to disburse the activity. With little modification in the networks configuration, most basic processes can be modified into environments. The process for the ADG was initially designed from the TEP network and then incrementally modified to fit the ADG process, reusing process networks and process objects.

CONCLUSION AND FUTURE WORK

As the Endeavors group investigates and builds workflows in other domains such as Web-based learning, workflow process components are likely to grow over time. The group is compiling these components into a library called E-lib (Endeavors Library), to allow rapid development of process workflow and to provide for reuse.

The SWAP protocol (Simple Workflow Access Protocol) and XML (Extensible Markup Language) allow interoperability and integration between other workflow tools over HTTP [4]. Since its development, WebNavigator has been designed to be SWAP compliant. We expect later versions to be fully compliant as the WebNavigator module and evolving SWAP protocol matures. XML will provide for better collaboration through semantically richer information between resources and workflow systems. SWAP, built on top of XML and HTTP, provides a partial solution for inter-workflow and application collaboration. A more specific and cohesive integration between applications (especially advanced workflows) will require a more descriptive interchange protocol. Endeavors currently supports reading, writing and distribution of XML encoded objects.

Chimera and WebDAV [12] provide support for WWW environments. Chimera is an open, serverized, hypermedia system that supports n-ary links between heterogeneous tools and applications in a network. We will use Chimera for managing resources, artifacts, and process activities to complement WebNavigator. WebDAV provides powerful extensions to HTTP for distributed authoring and versioning. As Endeavors objects are used throughout the WWW, distributed versioning will become increasingly difficult over time. We plan to leverage this technology to maintain consistencies of process objects over the Internet.

As the web matures, distributed components and web-resources will grow in size and demand. Distributed software engineering projects over the WWW, like Apache and Linux, are becoming more popular. Process tools for defining and deploying distributed development environments are useful for developing high quality software efficiently. Components and tools to support these workgroup activities are currently being developed to help automate the development process and others such as the specification, bug tracking, and testing.

TEP and ADG benefit from an open and extensible workflow process infrastructure combined with WWW technology. More importantly, this technology provides users and systems with the ability to create, invoke, and participate in process workflows over the WWW, implicitly or explicitly. Reusable components provide us with the tools to build distributed processes in less time and with less development effort so that developers can focus on building higher-level process components and programs in an interoperable and collaborative medium.

ACKNOWLEDGEMENTS
The authors would like to recognize Dick Taylor for his motivation and support, Peter Kammer for his review and input, and Greg Bolcer for his insight and vision. We would also like to acknowledge the rest of Endeavors team: Clay Cover, Ed Kraemer, and the Winter 1998 ICS 125b project class.

REFERENCES

1. G. A. Bolcer and R. N. Taylor. Endeavors: A process system integration infrastructure. In 4th International Software Process Workshop, pages 76-85, Brighton, U.K., IEEE Computer Society Press, December 1996.

2. R. T. Fielding, J. Gettys, J. C. Mogul, H. F. Nielsen, and T. Berners-Lee. Hypertext transfer protocol -- HTTP/1.1. RFC 2068, UC Irvine, DEC, MIT/LCS, January 1997. http://www.ics.uci.edu/pub/ietf/http/rfc2068.txt

3. A. S. Hitomi, P. J. Kammer, G. A. Bolcer, R. N. Taylor. Distributed Workflow using HTTP: Example using Software Pre-requirements. Proceedings of the 20th International Conference in Software Engineering, April 1998

4. P. J. Kammer, G. A. Bolcer, R. N. Taylor, A. S. Hitomi. Supporting Distributed Workflow using HTTP. Proceeding of the 5th International Conference in Process, May 1998

5. M. Kantor, B. Zimmermann, and D. Redmiles. From Group Memory to Project Awareness Through Use of the Knowledge Depot. Proceedings of the California Software Symposium, November 1997

6. Marimba, Castanet 3.0, http://www.marimba.com/

7. Netscape Communication, Netscape Enterprise Server, http://www.netscape.com/

8. Pointcast, Intranet Broadcast Tool, http://www.pointcast.com/

9. Sun Microsystems. Java Web Server. Available from http://jserv.javasoft.com/

10. R. N. Taylor, Dynamic, Invisible, and on the Web, University of California Irvine, Research Direction in Process Technology Workshop, Nancy, France, July 1997

11. R. N. Taylor and et al. A component and message-based architectural style for GUI software. IEEE Transactions on Software Engineering, June 1996, vol.22, no.6, 390-406

12. E. J. Whitehead. World wide web distributed authoring and versioning (WebDAV): An Introduction. ACM StandardView, 5(1): 3-8, (March 1997).

13. P. Young. Customizable Process Specification for Technical and Non-technical Users. Ph.D. thesis, University of California, Irvine. August 1995.