

Endeavors

Process Support System



Features

System enables definition, execution, and dynamic modification of workflows involving multiple artifacts, activities, and resources

Supports distribution of people, processes, behaviors, and artifacts

Customizable at all levels of abstraction

Flexible execution model Any mix of enforcement, guidance, and automation

Multiple complementary views/abstractions of processes and process state. Workflows may have any mixture of manual, partially or fully automated steps

Benefits

Improved management visibility into process definition and process state

Improved communication and coordination among team members. Facilitates knowledge of work assignment context and hand-off of work products

Automation of non-creative tasks

Standard computer-based office tools integrated with workflow/process tools

Better opportunity for measurement of process activities and optimization based on empirical data

Technical and non-technical workers able to perform job appropriate customizations of processes

Endeavors is an open distributed extensible workflow process execution environment

This is a screen shot of a requirements engineering workflow in Endeavors. The leftmost screen shows an end user view of a particular process activity; stakeholders interact through a common Internet browser. The two windows to the right of it are activity networks. The bottom network shows one representation of the main process to which process managers may visually add, edit, remove, or initiate process activities, before or during runtime. The network above it is a sub-network representing the expansion of the Review activity. The control panel at top enables users to access major components such as the network editor and object browser.

Overview

Endeavors is an open, distributed, extensible process execution environment. It improves coordination and managerial control of development teams by allowing flexible definition, modeling, and execution of typical workflow applications. Endeavors combines a sophisticated process modeling language with features designed for easy customization by both technical and non-technical users. The activities, artifacts, and resources of a process can be visually manipulated by end users to fit their current work context, even across the WWW. Technical users can define behaviors as well as integrate applications currently being used in their work culture using a variety of programming languages. This multiple-stakeholder approach to process technology allows company, site, and project specific customization.

At Runtime...

Endeavors provides a summary of assignments to each team member and coordinates hand-off and creation of artifacts such as documents, reports, or source code. Endeavors execution can be easily augmented with email or project management software, as well as project specific views.

Endeavors is easily configured to support executing processes locally or across platforms using the WWW. Team members and managers can track execution progress and change project state using a standard WWW browser. Endeavors may also be customized to provide further automation such as automatic invocation of tools, automated forwarding of documents to reviewers, or interaction with metrics gathering tools.

Design

Endeavors employs several key design strategies:

- ▶ Maintaining several layered object models, implementing the architecture as a set of highly componentized, lightweight, concurrent elements
- ▶ Supporting event-based communication between layers and components, including user interface components
- ▶ Reflexively modeling components in order to keep a dynamically distributable, customizable internal model of Endeavors, allowing dynamic loading of objects, behaviors, and user interfaces across the Internet (see figure 1)
- ▶ Multiple language support

Key Features and Properties

Integration. Endeavors allows bidirectional communication between its internal objects and external tools, services, and objects through its open interfaces across all levels of the architecture. Implementation of object behaviors in multiple languages is supported. Endeavors' object model is designed to facilitate flexible integration with development or end user tools. For example, Endeavors' agenda tool and default mail system can be replaced with third party time-management and mail systems, or be attached to a process activity awaiting execution.

Customization and Reuse. Endeavors is implemented as a layered virtual machine and allows object-oriented extension of the architecture, interfaces, and data formats at each layer.

Distribution. Endeavors has customizable distribution and decentralization policies which provide support for transparently distributed people, artifacts, process objects, and execution behavior through handlers. In addition, Endeavors processes, as well as the means to visualize, edit, and execute them, are easily downloaded using current and evolving World Wide Web (WWW) protocols.

Dynamic Change. Endeavors allows late-binding of resources and dynamic changing of object fields, methods, and runtime behavior. Process interpreters are dynamically created as needed.

Distinguishing Characteristics

<i>Other workflow process models</i>	<i>Endeavors</i>
Assume fixed user model	Dynamic configurable user model with customizable views for technical and non-technical users
Require an all-or-nothing buy-in	Componentized system that provides for incremental adoption. Integration with existing work contexts and applications
Lack the execution level semantics to do optimization and analysis	Rich execution model providing access to current process state and process history
Are point solutions and lack flexibility, not only for a project or domain but over time	Broadly applicable approach with a focus on supporting dynamic processes that evolve over time
Providing only limited control policies which occlude reactive control and graceful exception handling	Configurable control policies for flexible support of varying execution models

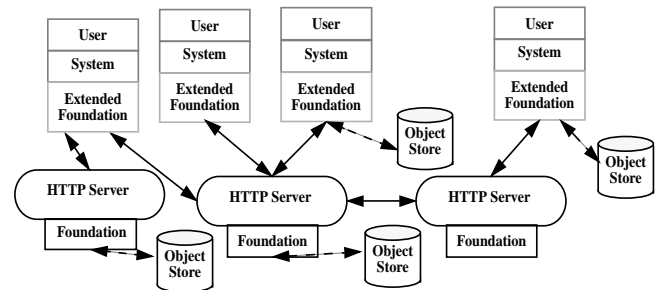


Figure 1

Contact Information

Professor Richard N. Taylor
 Professor David F. Redmiles
 Information and Computer Science
 University of California
 Irvine, California 92697-3425
 {taylor, redmiles}@ics.uci.edu
 949-824-{6429, 3823}
 949-824-1715 (fax)

For commercial companies wishing to obtain a commercial license please contact:

Dr. Gregory Alan Bolcer
 Endeavors Technology, Inc.
 12351 Ranchwood Rd.
 Santa Ana, CA 92705
 gbolcer@endtech.com
 714-505-4970

This material is based upon work sponsored by the Air Force Materiel Command, Air Force Research Laboratory, and the Defense Advanced Research Projects Agency under contract number F30602-97-2-0021. The content of the information does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred.