

Endeavors: A Process System Infrastructure

Arthur S. Hitomi

Gregory Alan Bolcer

Richard N. Taylor

Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92697 USA
+1 714 824 2704
{ahitomi, gbolcer, taylor}@ics.uci.edu

ABSTRACT

As software projects evolve, possibly differing in size, complexity, scope, and purpose, the development processes that support the project must evolve to reflect these changes. The Endeavors system is an open, distributed process modeling and execution infrastructure that addresses communication, coordination, and control issues. Endeavors' solution architecture applies five key design strategies: (1) maintaining multiple object model layers, (2) implementing the architecture as a set of highly componentized, lightweight, transportable, concurrent elements, (3) providing customization capabilities for each layer of the object model, (4) using a reflexive object model to support dynamic change, and (5) allowing dynamic loading and changing of objects.

Keywords

open, distributed process technology, architecture

INTRODUCTION

Effective software evolution demands that all stakeholders are able to participate in a system's evolution. This requires involvement in evolutionary 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. Endeavors is an open, distributed, extendible process system which has arisen from experience with the Teamware process modeling language [4] and lessons from a variety of other process technologies[2], software architecture and systems integration research. The design of the Endeavors system provides key mechanisms for support of distribution of processes and users, integration of third party tools, incremental adoption of the technology (leading to a low cost-of-usage entry barrier), customization and reuse of the process system components and process objects, and support for dynamic change of types and behaviors.

*LEAVE BLANK THE LAST 2.5 cm (1")
OF THE LEFT COLUMN ON THE FIRST PAGE
FOR THE COPYRIGHT NOTICE.*

*(preserve these six lines in some
cases, but make their contents
blank in your text)*

PROCESS ARCHITECTURE

Endeavors allows the object-oriented definition and specialization of the activities, artifacts, and resources associated with a software development process. The specification of processes and objects can be defined hierarchically; an activity can be further sub-divided into sub-activities, or a software development team resource can be further defined to include team member or sub-team resources. Endeavors' activity networks define the inter-relationships between activities, artifacts, and resources as well as sub-networks. Networks include the definition of control flow, data flow, and resource assignments, and can be easily defined using the graphical network editor. Activity networks can be executed by an interpreter which sends events to particular objects, triggering handlers which define the object's behavior.

Distribution

Endeavors has customizable distribution and decentralization policies which provide support for transparently distributed people, artifacts, process objects, and execution behavior (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 [1].

Integration

Endeavors allows bi-directional communication between its internal objects and external tools, objects, and services through its open interfaces across all levels of the architecture. Implementation of object behaviors in multiple languages is supported, allowing them to be described in whatever programming language is most suitable for integration.

Incremental Adoption

Endeavors requires low cost and effort to install across all software platforms. All process objects are file (ASCII) based allowing greater portability across different machine architectures. Components of the system, including user interfaces, interpreter, and editing tools, may be downloaded as needed, and no explicit system installation is required to view and execute a workflow-style process.

Customization and Reuse

Endeavors is implemented as a layered virtual machines architecture, and allows object-oriented extension of the architecture, interfaces, and data formats at each layer. Because processes, objects, tool integrations, and policies

can be used across platforms, processes may be adapted or evolved through embedding and composition of process fragments using cutting, copying, and pasting of activity representations.

Dynamic Change

Endeavors allows dynamic changing of object fields and methods, the ability to dynamically change the object behaviors at runtime, and late-binding of resources needed to execute a workflow process. Process interpreters are dynamically created as needed.

In order to achieve these goals, Endeavors employs the key design strategies of (1) maintaining several layered object models, (2) implementing the architecture as a set of highly componentized, lightweight, concurrent elements, (3) providing customization tools for each conceptual layer of the system, (4) supporting event-based communication between layers and components including user interface components, (5) reflexively modeling these components in order to keep a dynamically, distributable, customizable internal model of itself, and (6) allowing dynamic loading of objects, behaviors, and user interfaces across the Internet.

CURRENT STATUS

Endeavors is freely distributable software written completely in Java. Initial tests show the Endeavors system, interfaces, objects, and handlers interoperate and are portable across SPARC/Solaris 2.3, 2.4, or 2.5 and Windows NT/95. Eventually, Endeavors will be available for a variety of other operating systems. Endeavors may be executed locally as a standalone application or across the WWW. The current released version also includes:

- Installation and customization activities for populating the system with artifact and resources, based on users and installed software.
- Potential for automatic generation of WWW guidance page and support for third party project management tool (such as MS. Project 4.0).
- Support for object handlers written in Java and Python, with planned support for Ada95, JavaScript, and TCL scripts.

As the system allows incremental adoption, we have made various interfaces available for execution from the WWW and will continue to do so as they become available. Users can download and execute a demonstration process and the various system user interfaces using a Java-enabled web browser from our WWW page [6]. For the final release, the Endeavors installation, bug reporting, and customization processes will be able to be executed from this page as well as support for exchanging processes in a marketplace setting [3].

FUTURE

Future process work at U.C. Irvine will focus on extending the Endeavors system to better support distributed and post-deployment evolution of software by leveraging the contributions of our UCI partners in the Hyperware and Architecture research groups, as well as process researchers

elsewhere in the community.

Endeavors will support 1) hypermedia links between activities, artifacts, and resources, 2) innovative and domain specific interfaces including, for example, VRML worlds, 3) evolving WWW, http, and hyperweb protocols, and 4) software agent mechanisms based on a process representation. Endeavors will integrate and provide tools for rapid prototyping of WWW based process and guidance frameworks that are easily reconfigurable to support various work contexts and cultures. Endeavors will provide an infrastructure to encourage reusability of process fragments leading to a process marketplace by increasing customization capabilities to allow applying processes to new projects. This includes customization and experimentation with different interpretation, control, and transaction mechanisms.

ACKNOWLEDGMENTS

The authors would like to recognize the hard work and effort of the design and implementation of this system by Patrick Young, Peyman Oreizy, Peter Kammer, and Clay Cover. In addition we would like to acknowledge the members of the Chiron-2, Chimera, and WebSoft projects at UCI for their exchange of ideas during the development of this system. Finally we would like to thank Nenad Medvidovic and Kari Nies for their technical support and reviewing this paper

REFERENCES

1. Fielding, Roy and et al., "Hypertext Transfer Protocol - http/1.1", Internet Engineering Task Force (IETF), draft document, April 23, 1996.
2. Kaiser, Gail, Popovich, Steven, and Dossick, Stephen, "An extensible process server component and its integration into heterogeneous process-centered environments", SP96 at Brighton, England, Dec. 1996
3. Whitehead, E. James and et al., "Software Architecture: Foundation of a Software Component Marketplace", Proceedings of the First International Workshop on Architectures for Software Systems in cooperation with ICSE-17, Seattle, Washington, April 24-25, 1995.
4. Young, Patrick S, and Richard N. Taylor, "Process Programming Languages: Issues and Approaches", 17th International Conference on Software Engineering, Workshop on Software Engineering and Programming Languages, Seattle, Washington, April 23-30, 1995.
5. Endeavors, <<http://www.ics.uci.edu/pub/endeavors/>>
6. Endeavors demo, <<http://www.ics.uci.edu/pub/endeavors/demo/>>.
7. EDCS, <<http://www.ics.uci.edu/pub/edcs/>>

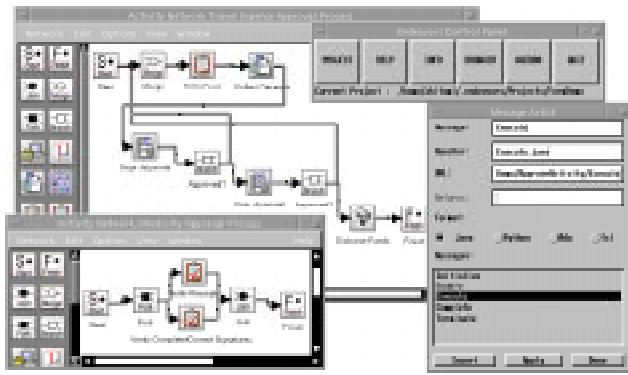


Figure 1. Screen shot of Endeavors Networks, Message, and Control Panel Artists.

