

Building the Process: Component-Based Workflow Architectures in a Distributed World

Peter J. Kammer

Endeavors Technology Inc.
19700 Fairchild Ave.
Irvine, CA 92612
+1 949 833 2800
pkammer@endtech.com

INTRODUCTION

Extensive work has been done, across a number of fields, in modeling organizational processes. A tension has existed between attempts to provide rich expression of organizational activities and a desire to support work as it actually takes places. There is the strong suggestion, backed up by observation, that the specification of many work activities is ad-hoc, based on the context in which the activity occurs. Typically, these diversions from the “ideal” process have been described as exceptions. The exception is tolerated, the work instance modified, or the model itself modified to address the inconsistency. For much of work activity, however, exceptions are the rule [6]. The process model serves as a way of describing the goal, rather than a guide for execution. To address this, some systems attempt to use more loosely coupled rule or event based approaches to model processes, but these tend to lack mechanisms for abstraction and describing well understood procedures.

An added challenge is the increasing reality of a widely distributed and dynamic environment. Process enaction may spread across any number of networked locations, even to mobile intermittently connected devices. Work processes stretch, not just across groups and sites within an organization, but across organizational boundaries. Relationships are fluid and may be created and optimized on an as-needed basis. Traditional centralized solutions lack the scalability and flexibility to approach these problems.

In the following sections, we present some of the challenges we see for dynamic workflow, and then discuss one potential approach combining a traditional procedural view of process with event-based distributed components to compose a process architecture.

CHALLENGES

Appropriate Levels and Types of Specification

Abstractions are an important element in breaking down process representation for understandability as well as providing a mechanism for dividing process ownership. These mechanisms are increasingly important in dynamic workflow as varying levels of abstraction should be supported to provide differing levels of specification where appropriate. This might depend on the how well the process is understood or the degree of variance in its enaction. Different elements of a process may lend themselves to different degrees of formalization and automation (e.g., [3]).

Modeling Dynamic Relationships Between Entities

Dynamism does not just reflect dynamic flow within a

process, but also dynamic relationships between the elements of the workflow. The agents (be they human or automated) of process enaction may change, as may the artifacts and resources available while the workflow is being enacted. Representing these relationships and the structure of their interactions can provide a richer view of process execution and a better understanding of the “architecture” of the workflow.

Distributing Workflow Specification and Enaction

Increasing amounts of business activity and the building of business relationships across the Internet involves dynamic work relationships that cross group and organizational boundaries and span the network. The description of work practice as well as its execution may take place at multiple locations with ownership of the overall business process split across organizations. This suggests a host of issues, including how to integrate work processes while concealing proprietary details, how to negotiate these dynamic relationships, what standards to use to communicate, and how to coordinate interactions and hand-offs of process execution from site to site.

Meaningful Visualization and Specification Mechanisms for Users

A key element of dynamically constructed processes is the increased need for human interaction with the system. Proper understanding of the current state of the work in progress, the resources available, and the activities taking places is critical in defining the next steps. Appropriate views will vary from user to user, depending on level of expertise, point of participation, and access. Defining the next step and activity is a collaborative process requiring mechanisms to cooperatively visualize and define the work taking place.

FLEXIBLE DISTRIBUTED PROCESS ARCHITECTURE

In order to support dynamic distributed processes, we propose an approach that borrows notions of architecture from software engineering. Workflow involves complex interrelationships between participants, artifacts, resources, and tasks. These interactions can be modeled in an event-based infrastructure that also integrates traditional notions of process flow. For our approach, we combine elements of the C2 software architecture style [7] and an object oriented model of activities, artifacts, resources, and agents derived from our earlier work on the Endeavors process support system [2].

The C2 architectural style describes software architecture in terms of software components and connectors. Components do not communicate with each other, but through events passed to connectors. Connectors and components are layered in a virtual machine architecture, providing access to events at varying levels of abstraction. An example of this approach is depicted, on a small scale, in Figure 1.

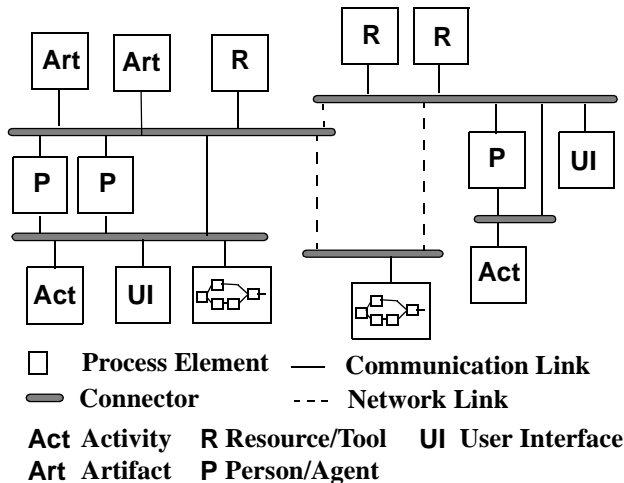


Figure 1. Distributed Process Architecture

Components in the architecture represent process elements as well as integrated tools and user-interface elements, interacting through events. Traditional procedural processes, modeled as components (depicted by components with activity graphs in Figure 1) may also be integrated. Distributed processes are interconnected through network links between their connectors.

Combining Reactive and Procedural Aspects of Process

The relationship between well structured procedural elements and weakly structure reactive elements is not necessarily one of direct hand-off. Rather, there is likely to be a ongoing interaction between procedural and reactive components of the process. By representing the traditional procedural process as one or more components in the event-based architecture, we may specify the interaction between it and reactive process components.

Supporting Dynamic Process Composition

The event based mechanism offered by this architectural style permits process components to be loosely or tightly coupled, depending on the demands of the process. Events may be synchronous or asynchronous and targeted to specific components or not. Process elements may be added to the architecture at run-time, allowing resources, agents, and artifacts to be chosen on the fly and respond to requests as needed. This process architecture also allows the communication and integration of independently executing processes so that access to a single resource or a single participant by multiple processes may be represented rather than modeling it independently in each process.

Representing Process at Varying Levels of Abstraction

The layered event model of this architecture provides the mechanism for aggregating events and representing process at appropriate levels of abstraction for different users. While low level activities may change and occur in a less controlled

manner, the status of those activities, reported through events, may be aggregated and applied to the process model at a higher level of abstraction, describing the progress of the process (more or less) in terms of its idealized form. An appropriate level of visualization for individual users may be obtained by “tapping in” at the applicable connector.

Supporting Distributed Process Execution

An important aspect of this approach is that it provides a mechanism for breaking up the architecture of process execution across multiple distributed sites. Processes interact across connectors. Network links between connectors distribute events to remote locations. Process components may migrate across the network, attach themselves to appropriate connectors, and access local resources (subject, of course, to access controls).

Encapsulating Process and Dividing Process Ownership

Ownership and execution of process often spreads across groups within organizations or even across organizational boundaries. Connectors provide a mechanism for both filtering and providing access to process events at an appropriate level of abstraction for external observers. This allows process owners to specify the level of visibility of their process. One internet site could, for example, compose a very large-scale process architecture by interconnecting the exposed connectors of various other sites.

Building on a Web Based Infrastructure

Development of this distributed process architecture is being based on an infrastructure provided by the open source MAGI system [1]. MAGI provides a personal mobile WEBDAV [8] server that includes an HTTP based event mechanism for communication, providing peer-to-peer collaboration mechanism for distributed users.

REFERENCES

- [1] G. A. Bolcer, “Magi: An architecture for mobile and disconnected workflow,” *Internet Computing*, vol. 4, no. 3, May/June 2000, 46-54. (see also <http://magi.endeavors.org>)
- [2] G. A. Bolcer and R. N. Taylor, “Endeavors: a process system integration infrastructure,” in *Proceedings, Fourth International Conference on the Software Process. Improvement and Practice*, Brighton, UK, 2-6 Dec., 1996, 76-89.
- [3] R. E. Grinter, “Doing software development: occasions for automation and formalisation,” in *Proceedings of ECSCW '97: 5th European Conference on Computer Supported Cooperative Work*, Lancaster, UK, 7-11 Sept., 1997, 173-88.
- [4] P. J. Kammer, G. A. Bolcer, R. N. Taylor, A. S. Hitomi, and M. Bergman, “Techniques for supporting dynamic and adaptive workflow,” *Computer Supported Cooperative Work*, vol. 9, no. 3/4, August, 2000.
- [5] P. J. Kammer, “Supporting dynamic distributed work processes with a component and event based approach,” *22nd International Conf. on Software Engineering*, 2000, 710-712
- [6] D. M. Strong and S. M. Miller, “Exceptions and exception handling in computerized information processes,” *ACM Trans. on Info. Systems*, vol. 13, no. 2, Apr.1, 1995, 206-33.
- [7] R. N. Taylor, N. Medvidovic, K. M. Anderson, E. J. Whitehead, Jr., J. E. Robbins, K. A. Nies, P. Oreizy, and D. L. Dubrow, “A component- and message-based architectural style for GUI software,” *IEEE Transactions on Software Engineering*, vol. 22, no. 6, June, 1996, 390-406.
- [8] E. J. Whitehead, Jr. and M. Wiggins, “WebDAV: IETF standard for collaborative authoring on the Web,” *IEEE Internet Computing*, vol. 2, no. 5, Sept.-Oct., 1998, 34-40.

AUTHOR'S BACKGROUND

Peter Kammer is a Ph.D. candidate at the University of California, Irvine, where he contributed to work on the Endeavors process support system and related projects. This included an extensive survey of research communities interested in process modeling, part of which is described in [4]. His research focuses on describing dynamic and widely distributed processes using component based models and leveraging off existing lightweight Internet standards and technologies [5]. He is also a senior software engineer at Endeavors Technology Inc. where he works on the research and development of web-based, mobile, and workflow applications supporting workplace and personal collaboration.

QUESTIONS

What sort of user interface is useful in guiding process participants in developing the workflow as they go along? How can complex process information be obtrusively specified by or captured from end-users?

How can workflow research learn from other CSCW approaches to support collaborative discovery of work processes?

Are the same models we've traditionally used for describing work processes appropriate for constructing models on the fly?

Attempts at standards for communicating process specifications and changes (WfMC, SWAP, etc.) have met with only luke-warm interest. Distributed dynamic processes and ad-hoc relationships suggest an increased need for such standards. What form should such a standard take or are existing proposals still appropriate?

For supporting emergent processes, what elements of organizational structure and relationships can be represented and monitored to facilitate understanding of the emergent process?