Guest Editors’ Introduction

Santosh K. Shrivastava
and Stuart M. Wheater
Newcastle University

Workflow-Management Systems

Workflows automate procedures for passing documents, information, or tasks between participants, and it does so according to a defined set of rules created to achieve an overall business goal. A workflow-management system should direct, coordinate, and monitor the execution of tasks arranged to form workflows (visit http://www.workflowsoftware.com). Although such systems are essential to organizations that must automate their business processes, they have been around for quite a while. So why have a special track on workflow? And why in IEEE Concurrency?

Dealing with heterogeneity

To answer the first question, we have observed that organizations are increasingly using electronic means for conducting business and, therefore, must automate their business processes. These processes include customer-to-business interaction (such as product support or customer complaint handling) as well as interaction between businesses. This naturally leads to business processes that cross organizational boundaries and raise challenging problems (visit http://www.crossflow.org).

Imagine an electronic travel-booking application. Figure 1 presents an activity diagram—the business process—depicting temporal dependencies between four constituent applications (or tasks): TravelPlan, CreditCheck, Flights, and Tickets. The CreditCheck and Flights tasks execute concurrently but only start after the TravelPlan task terminates and supplies the necessary data. Consequently, these two tasks have dataflow dependencies on TravelPlan. The Tickets task can start only after Flights terminates and supplies the necessary data and CreditCheck terminates in an okay state. In this case, Tickets has a dataflow dependency on Flights and a notification dependency on CreditCheck.

Several organizations must be involved
in this travel-booking application (a customer organization, travel agency, credit card agency, and so forth), and each organization can have its own workflow system for carrying out its activities. One way to execute this application would be for the travel agency to provide the application description (workflow script), coordinate the overall execution, and execute its TravelPlan and Tickets tasks; its workflow system would invoke the CreditCheck and Flights tasks at other organizations.

Clearly, if organizations are going to cooperate, there must be a standard way of representing an application’s structure and sending and receiving work items. Currently, it is not possible to deal with heterogeneity among users, workflow types, and workflow-management systems. An organization with many different workflow systems faces similar difficulties.

**Workflow articles**

In “Enterprise-Wide Workflow Management,” Christoph Bussler discusses problems facing large enterprises that require interworking between heterogeneous workflow-management systems. Such large enterprises are beginning to face the same challenges that arise when deploying cross-organization workflows. Bussler observes that integrating workflow-management systems can be extremely difficult because of the large variety of workflow metamodels systems possess. He presents some basic integration alternatives for heterogeneous workflow-management systems and examines issues involved in constructing a workflow-management infrastructure that presents a more homogeneous view to end users by encapsulating heterogeneous workflow-management systems.

Standardization efforts are underway to enable interoperability between workflow systems. Mark-Thomas Schmidt, who has played a key role in such efforts, takes a look at the work of the Workflow Management Coalition and OMG in “The Evolution of Workflow Standards.” He examines the present status of workflow standards in the areas of reference models, process definition, and runtime, and he reviews the recently adopted jointFlow OMG Workflow Management Facility standard.

In a wide-ranging article, “Processes Driving the Networked Economy,” Amit Sheth and his colleagues examine the future of workflow-management systems. The key technologies identified are workflow design, analysis, enactment, interoperability, and adaptability. Sheth and his colleagues examine each of these key technologies in detail, reviews the state of the art, and points out open issues that must be addressed.

**Distributed objects**

We hope these articles intrigue *Concurrency* readers. The technology that underpins (or ought to underpin) workflow-management systems is distributed objects: a mainstream subject of *Concurrency*. Clearly, we are dealing with large-scale distributed systems across the Internet. Present-day workflow systems are not scalable, as their structure tends to be monolithic. Furthermore, they offer little support for building fault-tolerant applications. We need to work on making them distributed and dependable (visit http://www.newcastle.research.oe.org/c3ds).

We must take into account many factors when designing distributed and dependable workflow-management systems. First of all, most workflows are rarely built from scratch; rather, they are constructed out of existing applications. It should therefore be possible to compose a workflow out of component workflows in a uniform manner, irrespective of the languages in which the component workflows were written and of the host platforms’ operating systems. Workflow composition, however, must take into account individual site autonomy and privacy requirements.

Second, the resulting workflows can be very complex, containing many temporal and dataflow dependencies between their constituent workflows. However, constituent workflows must be scheduled to run respecting these dependencies, despite the possibility of intervening processor and network failures.

Third, executing such a workflow can take a long time to complete, and can contain long periods of inactivity (from minutes to months), often due to the constituent workflows requiring user interactions. It should be possible, therefore, to reconfigure a workflow application dynamically because, for example, machines fail, services move or are withdrawn, and user requirements change.

Finally, facilities are required for examining a workflow’s execution history (for example, to be able to settle disputes). So, we must maintain a durable audit trail, recording the interactions between component workflows. Taken together, these are challenging requirements. We hope future articles in this track will cover these issues in detail, and we invite readers to contribute.

**References**


**Santosh Shrivastava** is a professor of computing science at the University of Newcastle and the leader of the Distributed Systems Research Group. He is also the technical director of Arjuna Solutions Ltd. (http://www.arjuna.com), which he set up with his colleagues to commercialize the academic research results of his group. His group is well known as the developers of an innovative distributed transaction system (called Arjuna) and a dependable workflow system for the Internet. His research interests are in the areas of distributed systems, real-time systems, fault tolerance, and application of transaction and workflow technologies to e-commerce. He obtained his PhD in computer science from Cambridge. Contact him at santosh.shrivastava@ncl.ac.uk.

**Stuart Wheeler** is a senior research associate in the Department of Computing Science at Newcastle, where he is a member of the Arjuna group. He also acts as a senior scientist for Arjuna Solutions Ltd. (http://www.arjuna.com), with special responsibilities for workflow products. His research interests include distributed computing, reliable systems, workflow-management systems, and object-oriented computing and tools. He received his BSc in computing science and his PhD from the University of Newcastle upon Tyne. Contact him at stuart.wheeler@ncl.ac.uk.