

## Interview: Amit Sheth on workflow technology



**Sushil Jajodia**  
George Mason University  
Fairfax, Virginia  
jjajodia@gse.gmu.edu  
<http://lis.se.gmu.edu/~csis/faculty/jajodia.html>

Amit Sheth directs the Large Scale Distributed Information Systems Research Lab at the University of Georgia (<http://lids.cs.uga.edu>). He is an associate professor of computer science at UGA and the president of Infocosm (<http://infocosm.com>). Before joining academia, he was in industry R&D at Bellcore, Unisys, and Honeywell for nearly a decade. His research interests include multiparadigm transactional workflow (the Meteor project), management of heterogeneous digital data and semantic issues in global information systems (the InfoQuilt project), telemedicine and teleconsulting through coordination and collaboration, and electronic commerce.

### WHAT IS WORKFLOW AND WORKFLOW MANAGEMENT? WHY IS IT POPULAR?

There are many definitions of workflow, but a simple and effective one is, *workflow* (or *workflow process*) is a computer-assisted (or automated) organizational process. An organizational process is a collection of activities related to a specific commitment, adding value to a product or service of the organization.

Workflow management is the coordination, control, and communication of activities in a workflow. A *workflow management system* (WFMS) is a suite of software products that can enable management (definition, testing, analysis, animation, integration, enactment, monitoring, and tracking, and so on) of workflow processes. While there are workflow management systems that only support human (performed) activities, we are interested in those that support both human and automated activities (including database transactions and legacy applications).

I like to see workflow technology as a paradigm to support *enterprise integration and programming in the large*. This is in the context that activities may be developed independently and at different times, implemented using different technologies, and supported by het-

erogeneous information systems in distributed (enterprise-wide or globally distributed) computing environments. A comprehensive WFMS can be used as an application-independent, infrastructure-independent glue to stitch together the heterogeneous applications into a mesh of cooperating units for achieving a unified solution.

A key reason for the popularity of workflow technology is the confluence of management trends with computing and communication capabilities in the early '90s. Management trends include reinventing and revitalizing American corporations through right-sizing and business-process reengineering, and by increasing their focus on process, sometimes at the cost of data (with the exception of the interest in mining and warehousing). Technical factors include significant progress in communication and distributed-computing infrastructure technologies. Recently, the latter increasingly involves Web and component technologies (Corba, Enterprise JavaBeans, or COM+). In market size, few software market segments are bigger than that of workflow. In 1996, the workflow-technology market was expected to be larger than \$2 billion (including software, services, and application development); in 2000, this is estimated to total approximately \$7 billion, including \$2 billion in software tools (that is, WFMSs).

### HOW HAS WORKFLOW TECHNOLOGY MATURED, COMPARED TO, SAY, TRANSACTION PROCESSING OR TRANSACTION MANAGEMENT IN DATABASES?

Although there was some early work on office automation, the term *workflow* originated in the mid-80s, and a technological push occurred in the early '90s (especially between 1993 and 1995). In contrast, some of the pioneering work in database management and transactions occurred in the late '70s, so you can argue that work in workflow technology followed that done



in transaction management and transaction processing by about 10 to 15 years. Workflow technology, although still in its infancy, has received very broad acceptance, and for a class of relatively simple applications, the technology is already proving its worth. While the software cycle has been compressed significantly, the second generation of workflow products is just about getting ready to enter the market. Nevertheless, the second or third version of a WFMS demonstrates the level of maturity that was demonstrated by the fifth or sixth version of a DBMS (this is similar to how fast Java IDEs (integrated development environments) are maturing compared to C++ IDEs). Two organizations, the Workflow Management Coalition and Object Management Group (specifically its Workflow Facility subgroup), are addressing the issues of interoperability.

The issues on the research side, especially for developing solutions for complex, mission-critical, and interenterprise applications, are fuzzier because of the very broad and multidisciplinary nature of this area.

**ARE THERE SIGNIFICANT ISSUES THAT INVOLVE WORKFLOW AND DATA INTEGRITY? WHAT ARE THE CONCURRENCY-RELATED ISSUES IN WORKFLOW RESEARCH, AND HOW DO THE ISSUES RELATE TO DATABASE TRANSACTION MANAGEMENT?**

Transaction-management technology is particularly adept at controlling data integrity because of well-understood properties of transactions, and well-understood and generally accepted correctness criteria based on serializability. This worked fine to the extent where extended transactions experimented with selective, yet fairly constrained, extensions of transaction properties and correctness criteria. The issues were harder when dealing with management of multidatabase consistency or consistency of related data managed in different databases and manipulated by different transactions.

Similar to the issues with multidatabase consistency, management of data integrity is important to workflow when workflow activities make updates to different databases or involve participation

of different processing entities (entities that perform the tasks in a workflow, including humans and information systems). Also, viewing consistent data is often important too. For example, if different tasks in a workflow access customer information from the same or different databases, then it is usually expected that the customer information obtained by different tasks (and subsequently updated) are consistent. As in the past, the problem is harder if different databases are handled. Furthermore, a task may need to read customer information that is distributed across different databases. In the distributed-database world, we knew the issues involved, but with nontransactional tasks and processing entities that do not support appropriate interfaces (such as for XA-compliant transactions), addressing the issue of data consistency is a real challenge.

Concurrency control requirements may be expressed as intraworkflow, interworkflow, or independently, say, as a general invariant. An example of an intraworkflow requirement may be that two tasks may be done in any serial order but not concurrently. An example of an interworkflow requirement may be that all tasks in one workflow instance must follow the corresponding (similar) task in another workflow instance (in the multidatabase transactions, this has been called "one transaction executing in the wake of another"). The more interesting set of requirements is what we may call, for the lack of a better term, *independent requirements*.

For example, if two workflows represent medical treatment (so-called clinical pathways) of different diseases of the same patient in the hospital, then the patient should not be scheduled for a procedure or lab work at the same time by different workflows. The issue becomes particularly interesting when developing adaptive workflows. If the patient was

being treated for one disease (as managed by an instance of a workflow) and if a new previously unknown disease (a so-called comorbidity) is found, a new workflow instance would need to be triggered to support treatment of the second disease. The two workflow instances would now have a variety of interdependencies such as the scheduling one mentioned above. Often, a workflow designer may not be able to predict these a priori, or to do so may be quite complex and still be incomplete or subject to change.

**SO CAN WE USE DISTRIBUTED OR EXTENDED DATABASE-MANAGEMENT AND DISTRIBUTED-TRANSACTION-PROCESSING SOLUTIONS TO SOLVE THESE PROBLEMS?**

There have been several efforts to start with extended transaction models for modeling and, in fewer cases, for (prototypically) implementing workflows. In fact, my first effort was indeed an attempt to apply an extended transaction model to specify a telecommunications workflow.

Now, I believe that although concepts and techniques for transaction management are important, perhaps critical, in developing WFMSs with capabilities such as error handling and recovery, an extended transaction model as a whole is an inappropriate basis for a general-purpose workflow model. The differences are truly fundamental for at least two reasons: human involvement is integral in a workflow (and humans don't behave as an information system—a DBMS—can be made to behave), and workflow management is a multidisciplinary area.

**WHAT ARE THE FUNDAMENTAL CHALLENGES, KEY RESEARCH AREAS, OR TECHNICAL HURDLES?**

On the fundamental level, understanding of the correctness criteria is still lacking. Let us divide the correctness issue into two categories: syntactic and semantic. There are several problems even with the syntactical aspects. A workflow model involves typically several components: a workflow map, data being manipulated by workflows, the organizational model, etc. A workflow map, to

## Having a formal basis is usually critical with respect to specifying and proving the correctness of any models in distributed computing and information systems.

which many models restrict themselves, is the main component and consists of several subcomponents, including control flow involving states of the tasks and data flow, and constraints (for example, deadlines). Often the issue of correctness has been limited to simply the subset of control-flow issues that can be specified, such as reachability and termination. The semantic part is inherently hard and usually application-dependent. Correctness involving relationships between workflow-model components, such as map and organizational structure, can be seen as a semantic issue and is seldom studied.

Having a formal basis is usually critical with respect to specifying and proving the correctness of any models in distributed computing and information systems. In this context, we can make two interesting observations. Many research studies in workflow involved starting with a formal model or language, such as variants of Petri nets, first-order and temporal logic, state and activity charts, process algebra, and so on. In these cases, the challenge has been to avoid extending the model with features needed for a comprehensive workflow specification that is not directly supported by the formal basis with which you started. And without significant extensions, such models have a hard time being of practical value.

The second approach has been (as with most vendors and some research systems) to develop a workflow model and a specification language independently of any formal model or language. While these are rich in expressiveness, we find that only a small subset of models can be mapped into formal counterparts for subsequent reasoning on syntactic correctness. To a good extent, this trade-off is not unlike that which faces software validation. Workflows automate organizational processes—processes help achieve objectives and products or services of organizations—and it is hard to find a static organization that does not change or whose environment does not change. For these and numerous other reasons, workflows need to be adaptive and supported by reconfigurable and dynamic WFMSs. This is an area of fertile research, as we need to understand how to model changes

and react to them. All issues of workflow modeling, including modeling features, correctness criteria, enactment services, and so on, need to be studied. Several assumptions made in most systems become invalid in many situations, resulting in the need for adaptive workflows. Some of these assumptions are that

- various aspects of workflow specifications are known a priori to the workflow designer;
- the execution environment and all its components are known to the workflow designer;
- a WFMS's scheduler can determine all execution sequences;
- a WFMS handles both normal execution and exceptions; and
- failure types and alternatives to handle them are known a priori.

One of the most interesting and challenging facets of workflow technology is related to the involvement of human participants. This has numerous implications, both for humans and for the WFMSs. This and several other issues have been discussed by a multidisciplinary group of researchers in the final report from the National Science Foundation Workshop on Workflow and Process Automation in Information Systems (<http://ldis.cs.uga.edu/activities/NSF-workflow>).

### ANY FINAL THOUGHTS?

An interesting aspect of workflow technology is that much of the serious research in workflow started after early products to manage simple workflows were having success in the market. That is, the researchers have not had a significant lead on the products and market development, and the scope of significant interactions and collaborations between researchers, practitioners, and product developers is excellent. In other words, much of the

research can be derived from real-world requirements that are already observed, rather than from imagined ones.

But perhaps the most interesting aspect about workflow for me is its true multidisciplinary nature (see the NSF workshop mentioned above). For the first generation of workflow products, vendors in several areas (e-mail, document management, imaging, and groupware) stepped up to extend their products. Now, vendors in several other markets (the Internet, the Object Request Broker, DBMS, and transaction processing) are poised to join.

As with the mélange of vendors with different roots in the workflow marketplace, workflow research attracts interest and contributions from many traditional research areas, including database-management and information systems, distributed systems, software process and software engineering, computer-supported cooperative work, organizational science, qualitative and formal methods and modeling, management information systems, and human-computer interaction.

From a somewhat broader perspective, I expect that there will be increasing interaction and integration between three technologies that are currently independent: coordination or workflow, collaboration, and information management (covering heterogeneous digital media). Such a technology integration will occur due to the need to support many complex activities involving complex human-and-information-system interactions and exploitation of large and heterogeneous information.

At the same time, there are more fundamental research questions related to such a convergence. NSF's Knowledge and Distributed Intelligence initiative (<http://www.nsf.gov/od/opa/news/publicat/nsf9860/starr.htm>) seems to offer a context for investigating some of these. WORP (Workflow and Process, <http://ldis.cs.uga.edu/worp>), an NSF-sponsored resource and repository, and the new ACM conference on work activity coordination and collaboration (sponsored by four SIGs) are expected to provide an attractive forum for fostering more interaction.