Tutorial: A Survey and Critique of Advanced Transaction Models

C. Mohan
IBM Almaden Research Center, San Jose, CA 95120, USA
mohan@almaden.ibm.com

Outline The classical transaction concept has been widely adopted in academia and in industry. That transaction model guarantees the ACID properties - atomicity, consistency, isolation and durability. In the last few years, for nontraditional applications like CAD/CAM, CASE, collaborative editing, etc. that traditional model has been found to be inadequate. To address the unique requirements of such applications and also to model business processes (workflow) which involve executing multiple interrelated transactions potentially on behalf of multiple users, several advanced transaction models have been proposed. Some of the latter are: Nested Transactions, Sagas, ConTract model, Flex Transaction model, Split-Transactions, etc. While much of the work has been done in academic research projects, some industrial groups are also actively working on these topics. Unfortunately, currently commercially available products incorporate very few of these results. Application writers have had to deal with the recovery and other consequences of splitting long running activities into multiple transactions. Even the decade-old nested transaction model is supported only by a handful of products, mostly in a primitive form.

With relational DBMSs beginning to support two-phase commit, stored procedures, triggers and threads, the role of traditional transaction monitors is currently being hotly debated. Under these circumstances, support for advanced transaction models is something that the current transaction monitors might be extended to provide to enhance the monitors' attractiveness and utility. While the X/Open distributed transaction processing standard for ACID transactions has been widely supported, at this stage, no serious standardization efforts are in progress with respect to advanced transaction models.

In this tutorial, I propose to review some of the work on advanced transaction models and point out those issues that have not been adequately addressed. I hope to provide an industrial and research perspective with implementation and practicality goals also in mind. This tutorial is intended for database and transaction systems' designers, implementers and administrators, users with high performance, high availability, high concurrency and nontraditional application requirements, and researchers in industry and academia.

Several papers on this topic have been included in the following publications:

1. A. Elmagarmid (Ed.) **Database Transaction Models for Advanced Applications**, Morgan Kaufmann Publishers, 1992.
2. M. Hsu *Special Issue on Workflow and Extended Transaction Systems*, **Data Engineering**, Volume 16, Number 2, June 1993.
3. A. Elmagarmid (Ed.) *Special Issue on Unconventional Transaction Management*, **Data Engineering**, Volume 14, Number 1, June 1991.

Instructor C. Mohan is a Research Staff Member at the IBM Almaden Research Center since 1981. He is a designer and an implementer of R*, Starburst and DB2/6000. He is a consultant for numerous IBM database and transaction processing product groups. He is the primary inventor of the ARIES family of concurrency control and recovery algorithms, and the industry-standard Presumed Abort commit protocol. His research ideas are incorporated in the IBM products DB2, DB2/2, DB2/6000, SQL/DS, MQM/ESA and ADSM. Mohan has received 4 IBM Outstanding Innovation Awards, 2 Research Division Awards and the 8th Plateau Invention Achievement Award for his patent activities (10 issued and 17 pending). In 1992, Mohan was elected a member of the IBM Academy of Technology. Mohan was the PC Chair of the 2nd International Workshop on High Performance Transaction Systems and a PC Vice-Chair of the 10th International Conference on Data Engineering. He is an editor of VLDB Journal. Mohan received a PhD in Computer Science from University of Texas at Austin and a BTech from IIT Madras.