Abstract:

Computing hardware is now available at a price, performance, and reliability point such that enterprise computing tasks may be performed on "commodity" class machines. These economic benefits have been achieved through producing large numbers of systems from standardized components. The OLE Transaction Architecture defines a set of distributed Component Object Model (COM) interfaces that can be used to coordinate distributed transactions using standard two-phase commit protocols. One of the goals of this architecture is to support the development of commodity enterprise computing software. A crisply defined and unambiguous transaction coordination architecture is a foundation for advanced applications requiring distributed transaction support. These applications may be developed and deployed in large numbers, achieving the same economic benefits that standard component hardware has experienced.

In this talk, we describe the motivation for developing the OLE Transaction Architecture, present the programming model for transactions as objects, and describe the implementation of the first Transaction and Resource Managers based upon the OLE Transaction Architecture.