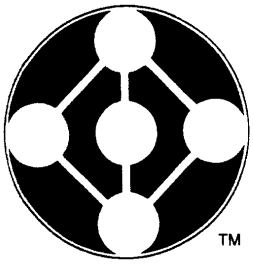

Alternative Technologies

Enterprise Integrity: Data and Process Consistency

Vol. 2, No. 3

Technology evolves haltingly, hesitantly, often making wrong turns and even going in circles. Such is clearly the case with integration technologies. The prevailing understanding of business process seems to have arisen from treating it as an extension of workflow between people-executed activities to the inclusion of software-executed activities. Similar thinking encourages treating message brokers as workflow managers. Two serious errors have resulted. First, it is commonly assumed that a workflow management system becomes a business process management system by enabling it communicate with application software. Second, few so-called business process automation systems differentiate between process (a.k.a. control) flow and data flow. Let's examine these errors, and their consequences, one at a time.

Traditional manual workflow is based on moving a set of related documents (such as a file folder) through a sequence of activities or work steps by passing the file to the people to whom those activities have been assigned. A cover sheet or interoffice mail folder may be used to specify the routing and to record task completion vis-à-vis signatures. Automating such a process electronically delivers an electronic version of the file folder and associated documents to the individuals assigned to the sequence of activities. Enhancements include automated scheduling, workload management across multiple resources that can handle a particular activity, email support, branching and joining, conditional branching, and so on. There may be some ability to logically group activities together. You probably knew all that.

Less well known are the limiting characteristics of traditional workflow, best understood by contrast with business process. A business process may consist of hundreds or even thousands of possible activities. By contrast, workflow has on the order of tens to a few hundred activities. Because the number of steps is small and the data handled is limited, it can make sense to move all the pertinent data through all the activities of a workflow even if not all the data is necessary at each step. Furthermore, workflow also typically has centralized control and is confined to a single business entity, while business processes can have distributed control under the influence of multiple business entities.

Business processes may have both forward and reverse flows of control (such as rework or message repair) and may permit conditional loops: workflow is essentially feed-forward. Perhaps most differentiating is the complexity of the unit of work and recovery from errors. A unit of work in a workflow is most likely a single activity, or perhaps a sequence of a few activities. If an error occurs, the work is routing to either an exception processing activity or through a special workflow that simulates reverse flow. Business processes are much more

complex and the unit of work may consist of many conditional steps, voting nodes, hierarchies of units of work, feed forward and feed backward flows, and so on. The result is that the sequential "roll-back" and "roll-forward" workflow technique of recovery often no longer succeeds.

Business process complexity, asynchronous message queuing, and the fact that not all a processes activities need all the cumulative data, insure that process flow is distinct from the data flow dictated by input/output requirements of individual activities. When process and data flows are tightly coupled and cannot be independently defined and executed, the business process management system will not be scalable. Data needed by any activity is moved to every activity and quite possibly through every route. (Aside: Question the scalability of a product if the vendor doesn't know how to spell the word: Its scalable, *not* scaleable and its scalability *not* scaleability.) This situation is made worse by our current obsession with the XML cure-all. Indiscriminate accumulation of data through many activities increases raw message size, and excessive formatting (due XML tags) support of message bloat, seriously threatening transformation performance and communication bandwidths.

Products which separate process flow from data flow, allowing separate sets of rules to be defined for each, will help mitigate message bloat. Just like dynamic changes to the process flow for workload balancing, dynamic data flow optimization would greatly aid scalability, flexibility, and reliability. But what are the rules by which we can move the least data over the shortest path while insuring that each activity has timely access to the data it needs when the business process management system schedules that activity? A little thought will convince you that this is a variant of the traveling salesman problem.

Hopefully I've convinced you not to equate workflow and process management when designing an enterprise integration solution. And don't give up correctness for performance or simplicity. It not worth it. Enterprise integration requires a self-consistent, coordinated confluence of data, process, and transaction. And only then does it ensure enterprise integrity.