Business Process Mashups?
Process Management and the Web Growing Together

Fabio Casati
University of Trento

Business process management, and more in general process and application integration technologies, have been around for over twenty years now. They have evolved from simple automation of the flow of forms and office documents among employees to full-fledged middleware tools that compose arbitrary applications in a secure and reliable (and even transactional) manner.

Despite the advancement in the technology, it is clear that process automation platforms have not lived up to their hype. Hundreds of models and tools have been developed and scores of researchers have devoted their time and energy to create models, theorems, algorithms, and prototypes for process management systems. Still, the pace of adoption of these technologies has been disappointing, and certainly not in the directions that were expected. Indeed, successful example of process technologies today rest mostly in specific areas, such as data extraction and transformation.

It is unfortunate to notice that, instead of trying to figure out why this technology is not as successful as we hoped (e.g., if there is some fundamental flaw in the underlying philosophy), we as researchers keep creating and extending models and algorithms, to a large extent seemingly devoted to improving travel reservation processes—by far the most popular application example for process and service composition tools, and apparently our biggest problem today.

The good news in all this is that the Web is coming to our rescue. Web 2.0 and in general novel Web applications are teaching us key problems we should look at, and roadmaps to solutions. Specifically, the Web has made people accustomed to having real time information on everything, and this will definitely apply to process monitoring as well. Real time information at a level of consumption that is meaningful for the various kinds of interested user will be a key features of process technologies, as important as automation itself. Another fundamental aspect is simplicity. People do not want to read user manuals or learn scores of complex languages with intricate semantics. And why would they. Anyways, they likely have to know java to create a process-based application (this is almost unavoidable), so why learn TWO complex languages? If composition languages are to succeed, they need to be simple, as simple as yahoo pipes, a composition language for RSS feeds and other Web content that can be used without any reading or any class, and that is providing us with interesting thoughts on where the complexity should reside and on the appropriate separation between complexity in the components and complexity in the composition language. Finally, I want to mention that, yes, the Web can learn from us as well. Mashups and user interface integration could certainly benefit from research in integration and middleware, such as event-based, publish-subscribe integration and even some of the research in service composition. The key once again will be to combine the rich functionality offered by “traditional” integration technology with the need for rapid and even end-user driven development characteristic of Web applications.