

Panel: The Impact of Research on Middleware Technology

Wolfgang Emmerich
Dept. of Computer Science
University College London, Gower Street
London, WC1E 6BT, UK

Mikio Aoyama
Nanzan University
27 Seirei
Seto 489-0863 Japan

Various commercial trends have led to an increasing demand for distributed systems. Distributed systems can integrate legacy components, thus preserving investment, they can decrease the time to market, they can be scalable and tolerant against failures. The caveat, however, is that the construction of a truly distributed systems is considerably more difficult than building a centralized or client/server system. This is because there are multiple points of failure in a distributed system, system components need to communicate with each other through a network, which complicates communication and opens the door for security attacks. *Middleware* has been devised in order to conceal these difficulties from application engineers as much as possible; Middleware is commonly defined as a software layer between applications and operating systems that provides application programmers with higher level of abstractions, such as remote procedure invocation, reliable message exchange or transactions. These abstractions considerably simplify distributed system construction and as a result middleware products are rapidly being adopted in industry and middleware is generally perceived as a success technology.

Though there is anecdotal evidence for middleware successes, it is very hard to quantify how successful middleware technology really is. An indicator might be how widely used middleware products are. Again those data are hard to obtain as the fact that someone has downloaded or installed an open source middleware or bought a license for a middleware product does not necessarily mean they are actively using it. However it is not necessarily required to have precise data here to get an impression and license revenue may serve as a useful approximation. A recent 2006 Gartner study determined that the world-wide market for middleware and application integration products has grown in 2005 to 8.5 billion US Dollars in annual license revenue.

When interpreting these numbers, it is worthwhile to bear in mind that the overall economic activity that critically depends on middleware is considerably larger for a number of reasons. Firstly, there are a significant number of deployments of open source middleware, such as JBoss and Jonas.

According to the Gartner study JBoss had 10 million installations in 2005 but JBoss is available free of charge and it therefore is not reflected in Gartner's study. Secondly, in addition to license revenues, middleware vendors have often significant revenues in professional services, which are not included here. Thirdly, the total significance of the application server market is larger yet as IBM, BEA, Oracle, Microsoft and Tibco rely on third-party integration service providers to deliver the bulk of their professional services and finally, the full economic significance of middleware can only be understood when we also take into account the wealth that distributed applications whose construction has been enabled with middleware create in domains such as e-commerce, finance, entertainment and education.

This panel will convene to discuss the impact of research on middleware technology, such as web services, application servers, message queues, distributed objects and remote procedure call mechanisms. With first-hand personal research and development experience on middleware, the panelists are well qualified to discuss this topic. Andrew Birrell worked at Xerox Park on the Cedar RPC mechanism, which is widely acknowledged to be the basis for all current RPC implementations and at DEC on Modula-3 Network objects that are recognized as the predecessor of Java Remote Method Invocation. Francisco Curbera is a co-author of the most important web services specification. Steven Reiss devised a broadcast message server for the Field environment, which he licensed to DEC where it became the foundation for one of the first commercial message queue implementations. Santosh Shrivastava's research in the Arjuna project led to the specification of the CORBA Object Transaction Service, which is to-date used as the foundation of the Java Transaction API in any J2EE application server. Andrew Watson worked on the definition and implementation ANSA at APM and later had a significant involvement in the standardization of CORBA as OMG Technical Director.