

## **Panel: The Future Of Distributed Object Computing**

Moderator: Bob Marcus - Director of Object Technology,  
American Management Systems

Panelists: Bob Atkinson - Microsoft OLE2/COM architect and designer  
Chris Horn - CEO and chief architect of Iona  
Hari Madduri - IBM DSOM chief architect

### **Introduction**

Each panelist will be given 20 minutes to outline his view of the future directions of distributed object computing. They will also be asked to address the issues of reliability, scalability and interoperability of future distributed object systems.

### **Hari Madduri (IBM)**

I believe the future of distributed object computing will contain the following:

- More robust and scalable object request broker implementations.
- Better interoperation among ORBs from different vendors
- Systems that can handle large volumes of transactions and objects.
- Object message queuing in addition to synchronous object calls.
- Objects distributed over large geographical regions and the use of intelligent agents.

In order for users to deploy mission-critical applications using distributed object technology, the fragility of ORB implementations is a concern. Also, practical considerations dictate that customers have to deploy objects on existing heterogeneous collections of machines. Therefore, out of necessity, more robust, scalable, and interoperable ORBs will be developed.

BIO: Hari Madduri received his M.S. and Ph.D.('85) degrees in Computer Science from the University of Wisconsin-Madison. He has worked on distributed systems, object oriented systems, and fault tolerance for over a decade. He was at Honeywell Research (Minneapolis) for five years working on various distributed systems projects. After joining IBM in April 1990 he worked on an advanced technology project aimed at building an object-oriented compound document toolkit (SOM grew out of this project) as an architect/designer. After that, he worked as the lead architect/designer of the Replication Framework component of the SOM Objects Toolkit. He is currently the lead designer and architect for the next release of DSOM.

### **Dr Chris Horn (IONA Technologies Ltd)**

Orbix is a full and complete implementation of the Object Management Group's (OMG's) Common Object Request Broker Architecture (CORBA). With Orbix, programmers can develop distributed applications using object-oriented client-server technology. Orbix provides a C++ language binding for CORBA. Orbix on Windows offers OLE integration.

Orbix today offers:

- broad platform base - including UNIX, Windows, Mac, Realtime and legacy operating systems.

- distributed OLE-CORBA and OLE-OLE communication today.
- scalability by using a light-weight ORB with services added as required.
- mission-critical reliability by layering on other products.

BIO: Chris is the CEO, Chairman and Chief Architect, IONA Technologies Ltd. He was a Professor of Computer Science at Dublin University, Trinity College until 1992, when he left to concentrate on IONA. He had ten years research experience in Distributed Systems, through European ESPRIT research projects. Chris holds an Engineering Degree and Phd from Trinity College Dublin.

### **Bob Atkinson (Microsoft)**

The broad-scale use of distributed computing is today quickly becoming a reality. But at the heart of the problem of how to support broad-scale distributed computing we find the age-old component software problem: what architectures and principles are needed to support the asynchronous development and deployments by independent design and development groups of software artifacts that interact with each other in rich and complex ways. There is, yes, more to the distributed computing problem than just the component problem (issues of network latency, for example) but it is our experience that the hardest and most difficult issues in distributed computing are in fact those that arise from the fact that using distributed computing in a broad scale is inherently a component software problem.

Once a well-designed architecture that successfully addresses the issues of the component problem exists, applying that architecture to solving the distributed computing problem should be natural and straightforward; indeed, if done carefully, the differences between using the component

architecture in an in-process environment can be made completely transparent to the implementors of both clients and servers. In contrast, tackling the design of a distributed computing infrastructure without the support of an underlying component architecture is fraught with pitfalls.

BIO: Bob Atkinson has had a keen professional interest in the issues surrounding software architectures and systems of independently developed components for many years. In recent times while at Microsoft he was one of the principal architects of OLE2 and its underlying Component Object Model, where he was responsible for the design of much of the underlying system architecture. Of late, he has been working in the area of transaction processing systems. He can be reached at [BobAtk@Microsoft.Com](mailto:BobAtk@Microsoft.Com).