PANEL: OOPSLA Distributed Object Management

John R. Rymer, *Seybold Office Computing Group, Moderator*
Richard Mark Soley, *Vice President, Object Management Group*
William Stephen Andreas, *Staff Software Engineer, Distributed Applications Architecture, Data General*
Ian Fuller, *Project Manager, Distributed NewWave Application Architecture, Hewlett-Packard*
Neal Jacobson, *Principal Engineer, Distributed Information Systems, Digital Equipment Corp.*
Richard A Demers, *Distributed Data Management Architecture, IBM*

SCOPE AND PURPOSE

Distributed object management has become a hot idea as vendors and users struggle to build increasingly complex information systems. Distributed object management is a framework for representing applications and services on a network through a common object schema. ALL entities within a distributed object management system are represented as objects, with their location on a network of heterogeneous systems being transparent to the user. Objects interact through standard interfaces, hiding their implementation details from other objects.

Distributed object management is not about object-oriented programming, nor is it about object-oriented DBMS. Rather, distributed object management seeks to provide a general object model that allows a range of users to create and manipulate objects to suit their particular needs in a secure, robust environment. Thus, in distributed object management, objects may represent simple entities such as text strings or entire applications, such as a document management system or process- scheduling system. Objects can contain nonobject-entities and encapsulate existing software. Objects can be created primarily by programmers or by end users or by agents, such as mail filtering agents.

Most systems vendors have made distributed object management a key part of their systems architectures strategies for the '90s and beyond. They believe distributed object management is a better way to represent and implement information systems in heterogeneous environments than current procedural methods. In their efforts to use distributed object management, however, these vendors have encountered difficult issues. They are:

NETWORK OBJECT MODELS

Object models can describe a diverse set of entities, and support development of both prototype and production objects. However, object models aren't sophisticated enough yet to accommodate the high degree of individual variation a distributed object environment introduces to objects. In distributed object systems, each user and environment may have a different implementation of a type. Also, each user or environment may require individual variations on the type itself.

SERVER-BASED AGENT TOOLS

Agents are a major component of distributed object management. Agents perform tasks for users, such as reading and sorting mail. Much research has gone into user-definable agents. However, "heavy-duty" server agents acting on behalf of multiple users are less well understood.

EXTENDED SECURITY MODEL

The distributed object management security model must distinguish between access that is controlled by a real user and access controlled by an agent acting on a user's behalf. The security model must also support more complex access methods than current models.

TRANSPARENT DISTRIBUTION

Hiding the location of objects is the goal of distributed object management. However, transparent distribution of objects across a complex network may overwhelm a network.

REPRESENTING TYPE DESCRIPTIONS

Distributed object management systems allow objects to be shared. However, existing ways of describing an object's type don't provide enough information to assure that two objects can interact. Types are defined by a list of an object's behaviors; without greater specificity, there's no way an object can really understand the capabilities of another object.

EXTENDED USER INTERFACE MODEL

The complexity of objects in a distributed object management system overwhelms today's graphical user interfaces. Complex objects, with large numbers of relationships, methods, and attributes outstrip today's approach of displaying object attributes in windows, methods in a menu bar, and services as "desktop" icons.

LACK OF STANDARDS

There are no standards governing the representations and interchange of objects. Also, it's not clear how existing standards like SQL will evolve to accommodate objects.

Many, many vendors and users believe distributed object management as a better model for representing and implementing information systems in heterogeneous environments. In this session, Digital, Hewlett-Packard, and Data General will discuss their efforts to use distributed object management in their systems architectures, highlighting the technical and business issues they've encountered. A representative of the Object Management Group will discuss the current state of standards in distributed object management. Lastly, an independent software vendor will discuss the opportunities it sees for using distributed object management techniques today and in advance of standards.

INDIVIDUAL PRESENTATIONS

The vendors active in distributed object management each are seeking solutions to these problems in their own ways. And the Object Management Group, an independent vendor-user consortium, is working to achieve consensus on standards. The panelists will describe these efforts, starting with the Object Management Group.

Richard Mark Soley, Object Management Group

Dr. Soley leads the Object Management Group's Technical Committee. The Technical Committee is completing a Standards Manual, which initially includes a statement about common object terminology and a reference model for distributed object management systems. Later versions of the Standards Manual will define standard interfaces between objects and services and object-requestor services in a distributed object management environment. Soley will describe the OMG's model and its implications for standards

Stephen Andreas, Data General

Data General has been crafting a distributed object management environment based on its Distributed Applications Architecture (DAA) for three years. It is a founding member of the Object Management Group. The first products based on DAA are due out in late 1990. They are expected to provide a network object server to support distributed object management for a variety of clients.

Andreas will describe Data General's efforts, including the issues it has encountered in network object models, object type representation, security, and user interface.

Ian Fuller, Hewlett-Packard

Hewlett-Packard's NewWave environment, released for DOS PCs in 1988, is built on an object management model, as implemented through an Object Management Facility. HP is also an OMG founder. HP is working on a Distributed NewWave Application Architecture for future versions of NewWave. The architecture will be implemented through a Distributed Object Management Facility.

Fuller will describe DNAA and its relationship to other object- oritented technologies, including object-oriented databases and object-oriented language systems. He will also discuss the issues HP faces as it implements its distributed object management strategy. These include: interoperability, standards, and acceptance by developers.

Neal Jacobson, Digital

Digital has been quietly architecting distributed object management services to be included in its Network Application Support (NAS) basket of application services. Digital's goal with these services is to make integration of heterogeneous applications easier to achieve. Digital previewed the capabilities it expects to include in its platform in DECwrite and DECdecision, two applications released last year. The two incorporate a hot-linking facility and direct manipulation of objects to create programs.

Jacobson will describe Digital's approach to modeling applications and the information they operate upon in an object- oriented way. Digital is a member of the OMG.

Richard A Demers, IBM Corp.

IBM's Distributed Data Management architecture defines object- oriented models of record and stream files and relational databases. Requests for operations on these objects can be directed to local files/databases or to those managed by remote systems. DDM products have been available since 1986 and DDM is now Common Communication Service of IBM's System Application Architecture.

Demers will describe the object-oriented aspects of DDM architecture and discuss how its object-oriented framework can be applied to a wide range of additional distributed object types.