

Message-Oriented Middleware

Introduction of MOM

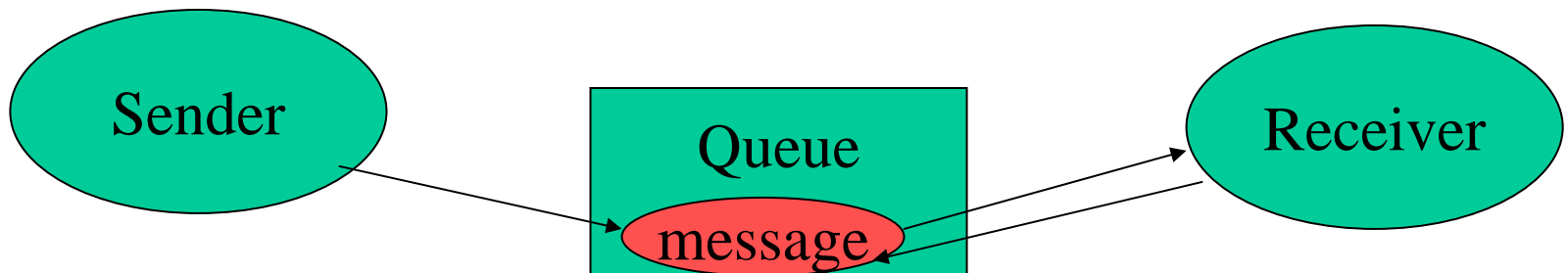
- Support the communication layer
 - also provide management layer function by vendors
- Main characteristics
 - a simple, flexible, and resilient form of interconnectivity
 - have a small and very easy-to-learn API
 - the level of flexibility and network transparency
 - programs can be added or removed easily
 - scheduling based on message priority level
 - applications to be designed to optimize the use of resources
 - can run in parallel
 - can run without waiting for replies
 - can even be deliberately delayed until a later time
- Advanced MOMs
 - mobile computing and ORBs

Messages

-
- Message
 - an arbitrarily long sequence of bytes that typically represents a ‘unit of meaning’ to the application processes that send and receive it
 - example
 - a video image, a DB update record, an e-mail message
 - carries information
 - carry action requests between applications or
 - carry action requests between processes in an application
 - be sent to just one other process
 - be sent to many specified processes, a defined sub-set of the service
 - multicast message
 - be broadcast to every user of the service
 - broadcast message

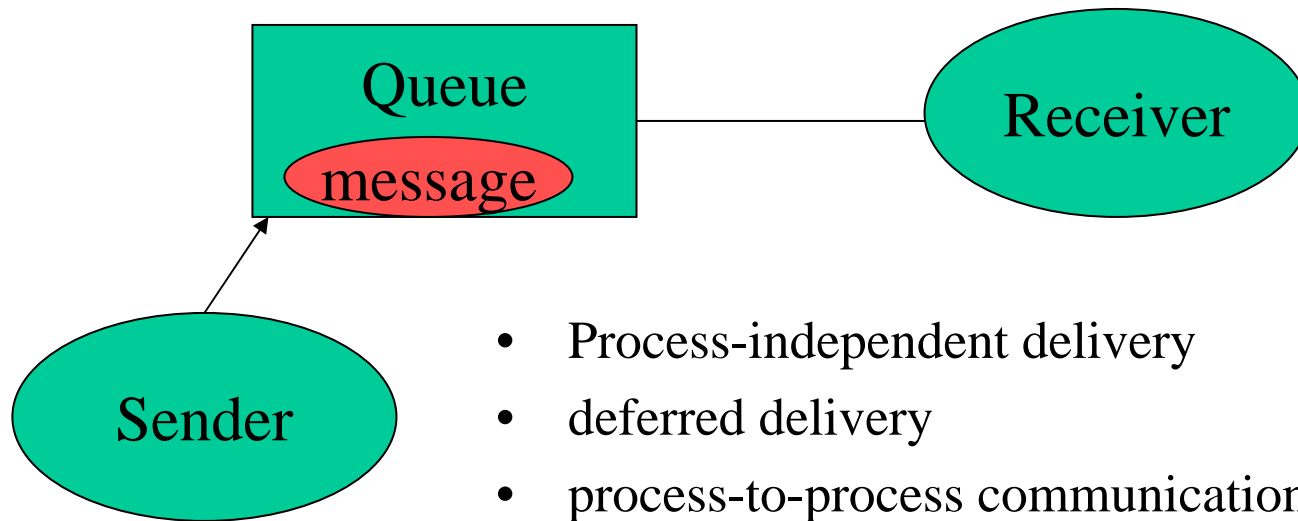
Store-and-forward (1)

- MOM
 - support asynchronous processing
 - communicating processes work independently and simultaneously
 - support synchronous transmission of messages
 - a message can be sent irrespective of the recipient's state of readiness
 - a message can be handled when it arrives
 - a store-and-forward
 - to handle inter-process communication at run-time



Store-and-forward (2)

- MOM (Cont.)
 - connection-independent
 - time-independent



- Process-independent delivery
- deferred delivery
- process-to-process communication

Services of MOM

- MOM SW
 - basic services
 - queue management and message passing
 - multi-protocol support
 - management-layer services
 - message routing
 - various delivery options
 - notification options
 - message prioritization
 - guaranteed delivery
 - system management functions
 - other services for reliable, scalable, performance
- Client-Server and Peer-to-Peer architecture

How do MOMs work? (1)

MOM

-
- Run-time services
 - message-queuing software
 - polling methods
 - Message Queue
 - PUT
 - GET
 - message-queue interface
 - Communication on different nodes or a single node
 - Types of communication supported
 - one-to-one communication
 - one-to-many communication: broadcast or multicast
 - many-to-on communication

How do MOMs work? (2)

-
- Shared queues
 - Deferred delivery and triggers
 - Sessions
 - session-oriented communication
 - session-less communication
 - Guaranteed delivery
 - Performance-related functions
 - Development environment

Moving from MOM to MOD (Message Oriented Databases)

Charled Brett

Middleware, 1997, 8, p. 10-19

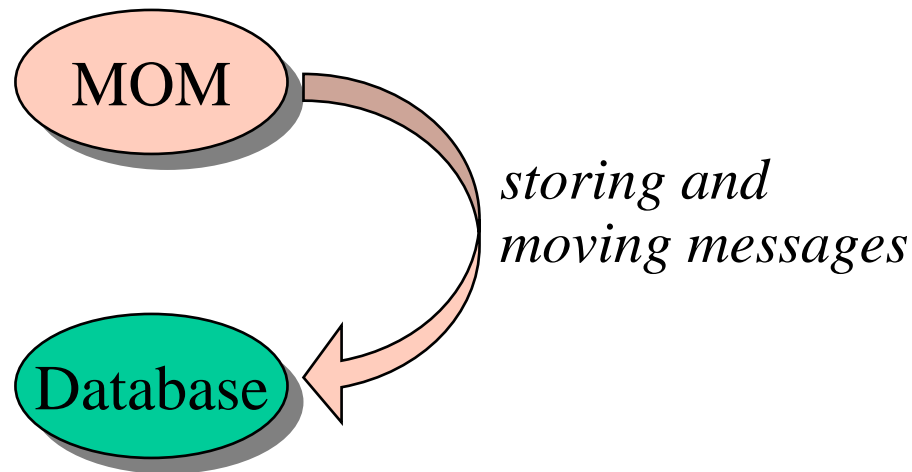
Contents

MOM

- Management introduction
- MOM and queuing
- The difficulty with the traditional MOM model
- Products
 - Sybase and *dbQueue*
 - Informix and the *Event DataBlade*
 - Computer Associates (CA) and *Jasmine*
 - Oracle and *AQ*
 - IBM (*MQSeries*) and Microsoft (*MSMQ*)
- Management conclusion

Management introduction

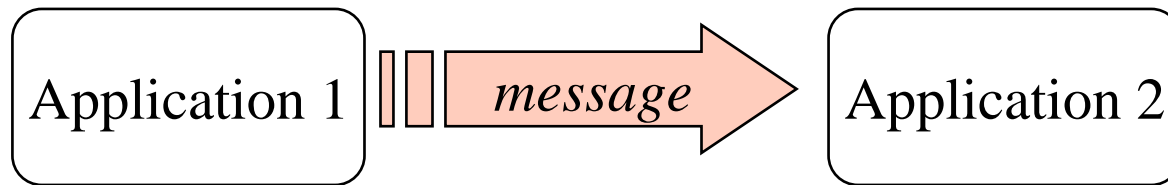
- MOD (Message Oriented Database)
 - a different proposition of MOM



- six vendors
 - Sybase (*dbQueue*) , Informix (*Event DataBlade*)
 - Computer Associates (*Jasmine*) Oracle (*AQ*)
 - IBM (*MQSeries*), Microsoft (*MSMQ*)

MOM and queuing

- Messaging system
 - essentially based upon some form of store and forward model



- decoupling of component parts in a distributed system
- Two underlying technologies of the broad store and forward model
 - queuing
 - propagation

MOM and queuing (cont.)

-
- Most messaging system
 - deliver messages from one node to another
 - storage and protection of messages
 - e.g., IBM's MQSeries, PeerLogic's Pipes,....
 - To deliver the necessary functionality for messaging to be reliable
 - storage mechanism
 - e.g., MQSeries : DB2
 - transfer agents
 - e.g., MQSeries : queue manager

MOM and queuing (cont.)

-
- ⇒ reliable unique messaging
 - ⇒ three separate “*transactions*”
 - an application placing a message on a queue
 - when the queue manager takes a message on Node A for passing to Node B
 - when the message on Node B’s queue is passed to a recipient application and confirmation is received that the message has been passed successfully
 - ⇒ “*transactionality*” : provides the assured processing between multiple platforms upon which distributed processing depend

The difficulty with the traditional MOM model

MOM

- Difficulty
 - implementation of the data stores for the messages
 - two different dimensions
 - the pragmatic
 - the technical
 - *solution!!*
 - Place queues in databases
- Advantages (MOD)
 - transactional consistency
 - ease of administration
 - reliability
 - performance

Sybase and *dbQueue*

-
- Sybase : first database vendor
 - a third party (NEON : New Era of Network) \Rightarrow dbQueue
 - A database queuing system
 - easy-to-use
 - reliable way to integrate applications across the internet, intranet, and client/server system
 - Features
 - integrate : message + Adaptive Server 11.0 database
 - cost reduced by eliminating the need for a separate message queuing product
 - simple : simplified messaging administration
 - reliable : guaranteed message delivery, security

Sybase and *dbQueue*

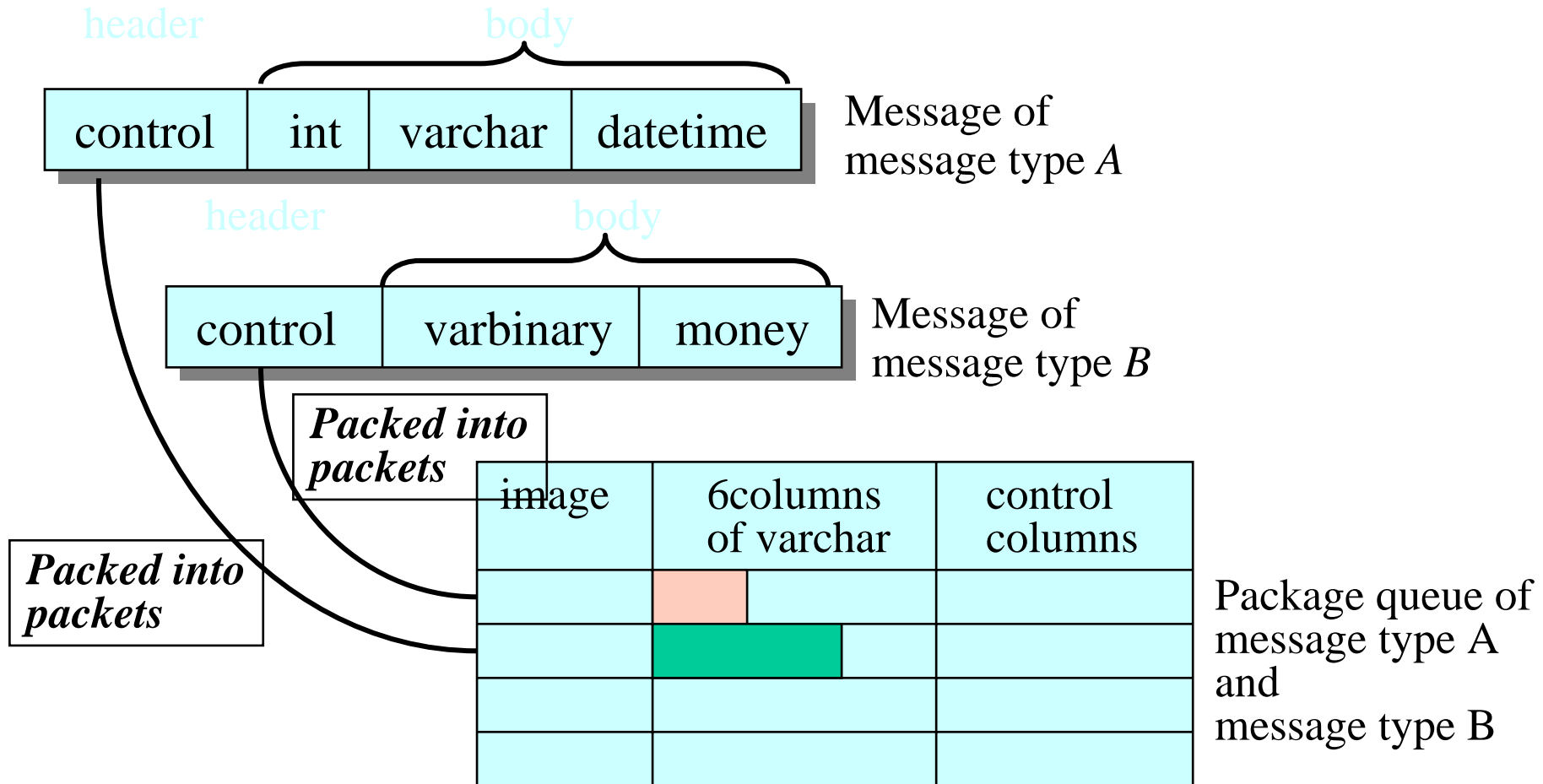
-
- Features (cont.)
 - high performance
 - highly optimized transfer protocol
 - performance tuning
 - integrated database and messaging operations
 - stores messages for applications until they are needed
 - queues are implemented as tables and stored procedures
 - UNIX platform, NT
 - API libraries
 - SQL, CT-Lib, ODBC, JDBC, C++Objects, ActiveX
 - Supported languages
 - SQL, C, C++, Visual Basic, Power Builder

Sybase and *dbQueue* (cont.)

-
- Queues are either visible or packaged
 - *a visible queue*
 - a database table, accept only one message type
 - each message is stored as a row in a table
 - *a packaged queue*
 - message data placed in packets, stored in variable size column
 - QTM (Queue Transfer Manager)
 - runs at the source queue node
 - continuously monitors source queues for undelivered messages
 - transfers messages to destination queue

Sybase and *dbQueue* (cont.)

- Visible vs. packaged queues



Informix and the *Event DataBlade*

-
- Database/DataBlade combination
 - integrates (Event DataBlade)
 - Informix database + TIBCO's publish/subscribe tech.
 - => SilverBlades
 - Features (extends the publish/subscribe model)
 - performance
 - easy-to-use
 - web-ready advantages : scalable, real-time
 - event-driven characteristics
 - real-time reporting and monitoring
 - real-time one-to-n data replication

Informix and the *Event DataBlade*

-
- Features (cont.)
 - storing and managing messages within the database
 - guaranteeing delivery of these messages
 - enabling users to define events in the context of business rules
 - allowing almost any type of information
 - Conjunction with database *triggers*
 - ⇒ used for workflow management
 - ⇒ provide the core for coordination of numerous tasks
 - starting tasks
 - monitoring tasks
 - invoking other tasks when appropriate
 - passing information between tasks

Informix and the *Event DataBlade* (cont.)

-
- Supports transactional publish/subscribe
 - publish / consume messages
 - update a local database atomically
 - After the transaction commits,
 - the Event DataBlade guarantees that published messages are delivered to all subscribers, in an asynchronous fashion
 - Publisher and subscriber applications
 - customer-written database applications
 - publish and consume from event tables in Universal Server

Informix and the *Event DataBlade* (cont.)

-
- *Publication agent*
 - is notified by the Event DataBlade
 - a process responsible for collecting all outbound messages and publishing them onto the TIB using TIB/Rendezvous Certified Messaging
 - *Listener agent*
 - working on behalf of local subscribers
 - receives the published events from Rendezvous
 - inserts these into local event tables
 - *Garbage collector daemon*
 - periodically checks event tables
 - to see if they contain any expired events that have been successfully delivered to all registered subscribers.

Informix and the *Event DataBlade* (cont.)

-
- *How it works*
 - publishing and subscribing applications communicate using TIB subjects outbound and inbound messages
 - For outbound communications,
 - the datablade application will declare a subject and set of message content using Informix procedural SQL
 - All outbound messages are stored in database tables
 - persistence and where desired, for automatic logging of all queuing activity
 - For inbound communications,
 - the listener process subscribes to the subjects list representing all subjects initialized by Informix database applications

Informix and the *Event DataBlade* (cont.)

-
- Advantages
 - core messaging functionality is exposed via standard SQL \Rightarrow be used by SQL-aware developers
 - guaranteed message distribution
 - enabling of decoupled and anonymous applications
 - asynchronous delivery of messages
 - scalable, reliable multi-cast dissemination of published events
 - message scalability
 - transactional handling of event publication and consumption
 - transactional messaging without two phase commits
 - location transparency
 - support for transient as well as persistent listeners
 - security

Computer Associates (CA) and *Jasmine*

MOM

- Jasmine
 - the first pure object solution
 - building next-generation business systems over Internet and client/server computing environments
 - support for
 - interactive multimedia
 - business transaction : electronic commerce
 - the speed and ease of use of the development environment
 - automatically and transparently handles the flow of data across the network
 - frequently used data is cached locally and the cache is automatically synchronized
 - infrequently used data is retrieved on demand

Computer Associates (CA) and *Jasmine* (cont.)

MOM

- *Object Delivery Manager*
 - optimize the transfer of data
 - only transmitting required fields while at the same time being capable of handling the high volumes inherent in multimedia data
 - transports data between its resource and a Jasmine client
- include access to information in
 - DB2, IDMS, VSAM files, OpenIngres, Oracle, Sybase

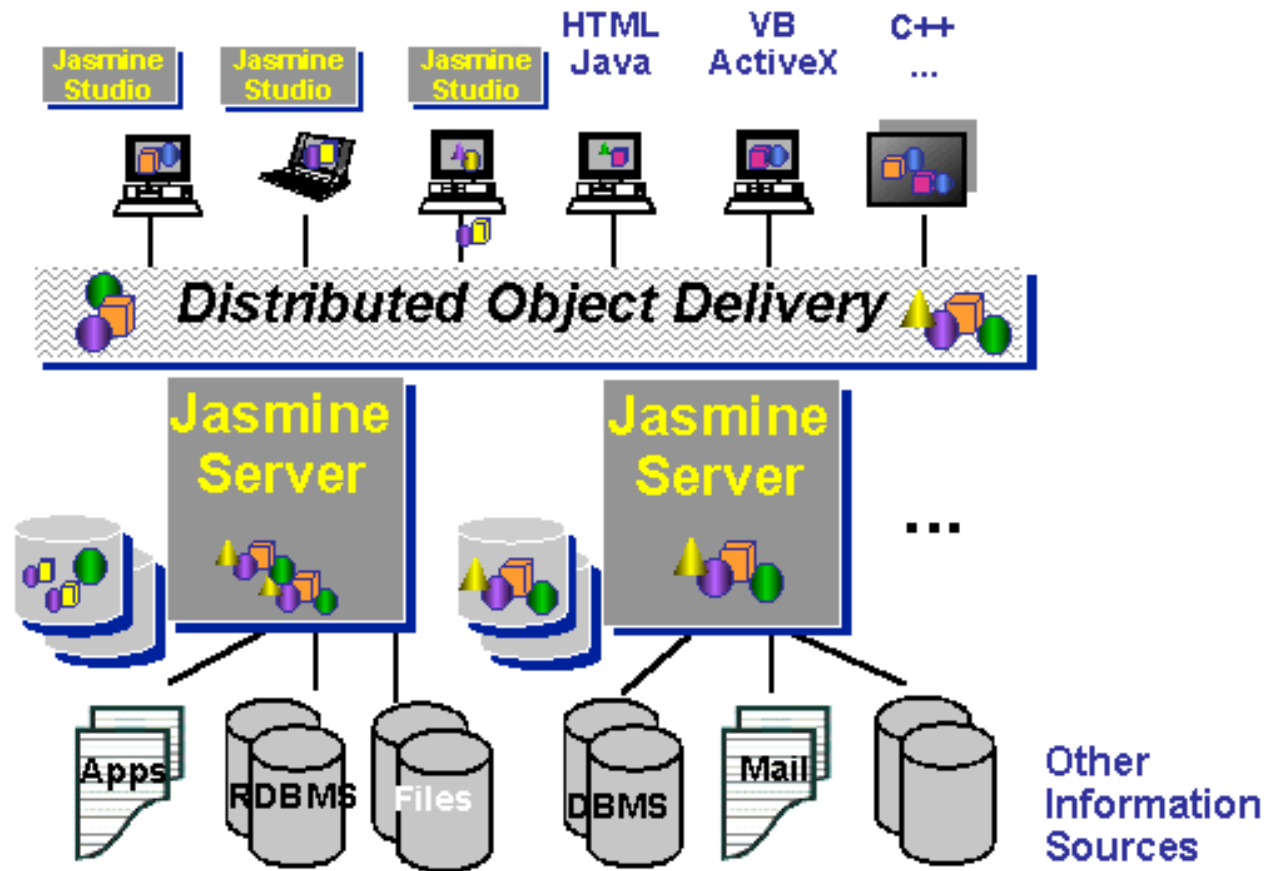
Computer Associates (CA) and *Jasmine* (cont.)

MOM

- Features
 - objects (or applications) on one or more computers
 - guaranteed message delivery
 - routing rules
 - support of multiple communications protocols
 - encryption
 - resilience
 - self-configuring development
 - scalability
 - a lightweight footprint

Computer Associates (CA) and *Jasmine* (cont.)

MOM



Oracle and AQ

-
- ORACLE AQ (Advanced Queue)
 - a message queuing system with the Oracle database
 - allows users to store messages into queues for deferred retrieval and processing by the Oracle Server
 - Functionality
 - persistent queuing
 - ability to specify a time interval for queue elements
 - ability to specify a time interval for queue elements
 - integral transactions
 - priority and ordering of queue elements
 - ability to dequeue multiple queue elements as a bundle
 - ability to specify multiple recipients

Oracle and AQ

-
- Four basic entities
 - *messages*
 - small unit of information being inserted into and retrieved from a queue
 - message = control information + payload data
 - *queues*
 - repository for messages
 - two types : user queues, exception queues
 - *queue table*
 - queues are stored in queue table
 - each queue table contains a default exception queue
 - *agents*
 - queue user
 - two types : producers, consumers

Oracle and AQ

-
- Features
 - integrated database level operational support
 - SQL access
 - windows of Execution
 - user can specify that the consumption of a message has to occur in a specific time window
 - multiple consumers per messages
 - navigation
 - user can select the first message or once user have selected a message and established a position
 - ordering messages
 - sort order, priority, sequence

Oracle and AQ (cont.)

-
- Features (cont.)
 - modes of dequeue
 - waiting for the arrival of messages
 - retries with delays
 - exception queues
 - message grouping
 - retention
 - message history
 - tracking
 - import / export

IBM (*MQSeries*) and Microsoft (*MSMQ*)

MOM

- In the marketplace,
 - *MSMQ*
 - high visibility in the NT and Windows arena
 - *MQSeries*
 - wide platform support (22+)
 - dominance on the mainframe
- Both MSMQ and MQSeries
 - as traditional MOM suppliers
 - queues live in specialized database
 - little intention of placing their queues and in SQL Server or DB2

IBM (*MQSeries*) and Microsoft (*MSMQ*)

-
- MSMQ
 - easy to integrate applications, implement a *push-style* business event delivery environment between applications
 - build reliable applications that work over unreliable but cost-effective networks
 - Features
 - easiest message queuing product to use, deploy, and administer on Windows NT and Windows 95
 - full COM support
 - dynamic directory service-based architecture
 - built-in centralized system management

IBM (*MQSeries*) and Microsoft (*MSMQ*)

-
- MSMQ
 - Features (cont.)
 - offers comprehensive message queuing functionality
 - reliable, resilient message delivery
 - cost-based message routing
 - full support for transactions
 - is fully integrated with other Windows NT features
 - MTS (Microsoft Transaction Server)
 - IIS (Microsoft Internet Information Server)
 - Windows NT clustering services
 - Windows NT security environment

Management conclusion

MOM

- MOD
 - enables users to re-use what they already possess
 - much easier to extend what you have already than it is to add whole new middleware environment
 - Even if MOD is not so efficient as MOM
 - it will likely be much easier to integrate

References

-
- Charled Brett, “*Moving from MOM to MOD (Message Oriented Databases)*”, Middleware, 1997, 8, p.10-19
 - <http://www.sybase.com:80/inc/corpinfo/press.246e.htm>
 - <http://www.silverrun.com/News/press/silverblades.html>
 - <http://www.tibco.com/press/press21.html>
 - <http://www.tibco.com/products/tibconnectinformix.htm>
 - <http://www.cai.com/products/jasmine/analysis/idc2/jasmine.html>
 - <http://www.cai.com/products/jasmine/analyst/aberdeen/body.htm>
 - http://www.datacraft.com/opp/dbms_aq.html
 - http://www.pori.tut.fi/~tim/kurssit/TiedonhallinnanSeminaari/Kasikirjoja/server803/A54643_01/ch_aq.htm
 - [http://www.microsoft.com/ntserver/nts/appservice/exec/overview/MSMQ_overview.a
sp](http://www.microsoft.com/ntserver/nts/appservice/exec/overview/MSMQ_overview.asp)