MQSeries and CICS Link for Lotus Notes*

Lotus Development Corp.
55 Cambridge Parkway
Cambridge, Mass. 02142
URL: http://www.lotus.com

1 Introduction

Every business today is investing in sophisticated and advanced computing architectures and platforms, exploiting robust databases and operating systems in a distributed, client/server network. Among the benefits provided by these new systems are improved access to data in usable form, superior price/performance ratios based upon scaleable server technology, ease of use based upon well designed graphical user interfaces and client software.

On the other hand, a large portion of the data that is critical to many business processes is stored and managed by transaction systems. Moreover, these systems manage high volumes of transactions – the bread and butter of any company. One of the most formidable challenges that Information Systems managers face is the need to marry new client/server systems with these transaction systems without sacrificing data integrity, performance or ease of use and without consuming extraordinary resources in expensive middleware and dedicated development and maintenance staff.

Current attempts to integrate modern systems and transaction systems represent a tradeoff that inevitably favor one characteristic (e.g., performance) over another (e.g., ease of use):

- **Terminal Emulation:** This solution requires the user to understand the syntax and become familiar with the user interface of a disparate system, creating a need for increased user training. In addition, the transfer of information usually requires some form of manual intervention, which introduces inefficiencies and inaccuracy. Also, the communications support for emulation of the older application must reside on the workstation.

- **Staged Data:** By making a query that retrieves and stores a large amount of data on the client/server system, users have direct access to data in a familiar and intuitive environment. However, this data immediately falls out of synch with the host system, and leads to data integrity and transaction integrity issues that can often present larger and more intractable problems.

- **Direct Access:** Direct access to data ensures that the client/server system is working with the most recent data and provides the user with a single point of reference. However, such access requires significant investment in middleware and application development. More

*MQSeries and CICS are trademarks of the IBM Corp. Lotus Notes is a trademark of the Lotus Development Corp., an IBM subsidiary. All other products mentioned in this paper are registered trademarks of their respective companies.

important, such data access ignores the larger issue of transaction integrity, sacrificing the security, recovery, logging and synchronization that are the hallmark of transaction systems.

Further complicating the integration effort is the use of products and solutions from multiple vendors, which often leads to integration, support and maintenance problems.

2 Requirements

A solution that effectively integrates transaction systems with new client/server systems needs to preserve and leverage the strengths of each platform with minimal tradeoffs in functionality and ease of use:

- **Support for Transaction Processing Monitor System:** Transaction processing applications rely on the system-level services provided by the transaction processing monitor, such as rollback, backup and recovery facilities, and logging and auditing functions. A complete integration solution needs to take advantage of this reliable infrastructure, and should not require the introduction of any additional infrastructure.

- **Transparency:** Requiring users to switch between two types of systems imposes a tedious and error prone process on end users. The integrated solution should present the user with a single, intuitive environment that is consistent with other client applications.

- **Reliability:** Information Systems managers, quality assurance staff and users depend on the completion of a transaction. The integrated solution must ensure that a transaction is processed, or alert the appropriate staff of failed transactions.

- **Cross Platform Support:** Client/server systems are characterized by a plethora of operating system platforms, including Windows, OS/2, Macintosh and UNIX. The integration solution must support the full breadth of client, server and host platforms.

- **Mobile Support:** It is important to note that many client/server applications make use of replication technology to support mobile and remote users. Mobile applications that presume periods of latency necessarily are fundamentally at odds with the real-time orientation of transaction systems. This conundrum adds a daunting level of complexity to bridging transaction and client/server systems.

- **Cost Effectiveness:** IS managers will balance the risk and cost of continuing with current, incomplete integration solutions with the cost of acquiring, installing and maintaining a new solution. A complete solution therefore needs to represent a compelling cost/benefit proposition.

- **Support for Standards:** The IS organization depends on the ability to invest in upgrades to systems software without introducing incompatibilities with existing systems. Therefore, an integration solution should be based on published standards and APIs that can be expected to be supported by future platform changes.

3 Notes and Transaction Systems

With millions of licenses deployed, Lotus Notes is one of the leading client/server systems today. Notes is an environment for the development and deployment of groupware applications. While these groupware applications are designed to manage unstructured data, many organizations have identified the opportunity to exploit Notes as a front end to host-based transaction processing systems. The benfit of extending the exisitng host transactions to Notes is to provide users with a single point of access to these systems and to client/server applications, as well as building applications that use both the structured data of transaction systems and the unstructured data of groupware.

In fact, for many users, Notes has become the computing "home" in which they spend the majority of their time. Notes groupware applications, client/server mail, integration with desktop productivity tools and its integration with third-party databases make it a robust, familiar environment in which to launch and run a wide array of business applications. Lotus and IBM's Hursley Lab have worked together to develop a technology that integrates Notes with transaction processing systems, allowing Notes to act as the "home" for transaction systems as well.

The technology, called IBM MQSeries link for Lotus Notes and CICS link for Lotus Notes, removes the constraints of partial solutions, providing an IS organization with a robust, transaction-oriented solution for integrating these systems with Notes. These links leverage existing transaction processing systems, requiring no changes in technology or business processes to conduct backup and recovery, logging and auditing, system measurement, workload balancing or performance monitoring. System security also remains unchanged. In addition, the links allow Notes to act as a common front end to a variety of otherwise standalone and incompatible systems. Also, the integration between transaction systems and Notes includes support for the mobile/remote Notes user.

4 MQSeries and CICS Link for Lotus Notes

MQSeries and CICS link for Lotus Notes are Notes server tasks that translate data between Notes and a set of APIs (MQI in the case of MQSeries and ECI for CICS). This link technology also controls the flow of data between the Notes application and the system transaction. The actual translation, connecting, delivery and reply from the target system are all under the control of the link technology. Note that a single user request from a Notes client can generate multiple requests to one or many target systems, allowing the work to be processed in parallel.

CICS link for Lotus Notes works directly with CICS, which runs natively on OS/2, MVS and VSE. A host based system with CICS requires no additional host software to integrate with Notes.

MQSeries link for Lotus Notes works with 16 back end systems that may not have support for CICS (see p. 7 for a complete list of platforms). Non-CICS systems require the MQSeries software on the host. MQSeries link provides messaging and queuing support, routing the message to the appropriate target system in the network so that it can be accessed by programs (e.g. a Lotus Notes server) servicing these queues. It avoids the use of private, logical connections between application programs, so that all communication occurs through queues only.

9 Managing the Link Process

The actual link and switching process is managed by controlling the MATM database. This is a normal Notes database using standard Notes conventions. It controls the link that is a Notes server Task, so its management is most appropriately performed by the Notes Administrator.

10 Platform Support

Today, the link supports both the OS/2 and AIX server editions of Lotus Notes and all Lotus Notes workstation clients. MQSeries link provides connectivity to DOS, Microsoft Windows; IBM OS/2, AIX, OS/400, MVS/ESA and VSE/ESA; AT&T GIS UNIX; DEC VAX VMS; Hewlett-Packard HP-UX; Santa Cruz Operation SCO UNIX, Sun Microsystems SunOS and Solaris; Tandem Guardian; and Novell UnixWare.

Messaging and queuing services are interconnected across various communication networks such as TCP/IP and SNA, providing application access to other applications anywhere in the network.

Lotus Notes supports all major operating systems including Apple Macintosh, IBM OS/2 Warp, Microsoft Windows and Windows NT, and UNIX platforms including IBM AIX, Sun Solaris, HP-UX, and SCO ODT. Notes is available as a NetWare loadable module for the Novell environment.

11 Future Enhancements

Lotus and the IBM Hursley Lab are working together to enhance the integration and functionality of MQSeries link and CICS link for Lotus Notes in 1996 as follows:

- **LotusScript Support:** This would extend the access of transaction systems to the entire Lotus suite of products. That is, developers would be able to build applications that not only integrate Notes with the transaction systems, but which also integrate Lotus desktop products, such as 1-2-3, with these transaction systems. In this way, a formula in a single 1-2-3 worksheet cell can trigger a request for information from a transaction processing system or actually initiate a transaction itself. In addition, this support would allow control of the Notes client for an application that requires steps to be completed sequentially rather than in parallel.
- **Host-initiated Batch Data Transfer:** The host transaction system will be able to initiate scheduled downloads of data to the Notes server.
- **Wider Support:** Additional support for other Notes server platforms by MQSeries would include Windows NT, HP-UX, and Sun Solaris.

7 Mobile Support

This asynchronous design also supports mobile and remote Notes users. A mobile user can replicate and carry the link data on a laptop computer . Since the links use a mail enabling technology, the requests to the legacy system would be stored in the link database. When the user replicates with the server all the link requests would be processed. The results are then either processed by directly returning them to the remote users or by storing them for the next replication. Mobile and remote users also have the option of making requests while connected to the server via telephone connection.

8 Application Development

The development of the actual application is not much different from any other Notes development effort. All the definition, design and testing take place in the Notes development environment.

To access transaction systems or non-Notes system data the Notes developer needs to know the key fields that are used in the host system to access the data and what fields they wish to have accessed. The linkage technology provides a programming interface for computers and networks from multiple vendors and offers a simple, reliable means of building distributed and client/server applications. The link shields the Notes application builder from the multi-vendor, multi-protocol complexity of today's business networks and provides application-location transparency.

For example, a Notes application is being developed to handle customer service. The basic demographic information (address, phone number, etc.) for that customer is stored in a transaction system and is keyed by that customer's name. The Notes application developer would define a field for the customer name, address, etc. During the definition process the developer mail-enables those fields or the form itself to either CICS link or MQSeries link. This means that the fields will be sent to the link when the form is saved or updated.

Where the application development process does differ from traditional Notes application development is in transaction mapping. In the transaction mapping definition, the developer creates a control record in the MATM database. This definition stage is somewhat similar to the establishment of a connection record within Notes. To define the transaction mapping, the developer needs the name, location, type and any security passwords, etc. for the transaction system; the name and key fields of the transaction application; what is being mailed to the link from the Notes application; and what is to be done with the results when they are returned. This "connection" record defines:

- The actual transaction system (e.g. CICS, IMS) or non-Notes system (e.g. OS/400, DEC VAX VMS).
- What Notes fields map to what fields on the defined system.
- Proper security.
- What actions are permitted on the data (adds, deletes, updates).
- How the results are processed.
- Any other fields needed for processing.

5 A Typical Transaction

A Notes application consists of a variety of objects, one of which is the Notes form, which in turn contains several types of fields. Some of these fields are populated with data obtained from a transaction processing system or other non-Notes systems. MQSeries link for Lotus Notes (or CICS link for Lotus Notes) provides an extension to a field definition describing how the data can be obtained or stored.

The Notes Administrator has set up the proper security, transaction identification and mapping to the target system. This key information is sent along with the field to a Notes database managed by MQSeries link. MQSeries link extracts the key along with the requested fields and switches these to an MQSeries request. MQSeries link then plugs in the security, transaction information (keys, fields, etc.) and issues the request to MQSeries. MQSeries then sends the request to the appropriate system using its routing technology.

When the transaction is complete, the requested data is placed in an MQSeries message and sent back to MQSeries link. MQSeries link searches for returned responses in addition to processing outgoing tasks. The incoming data and return codes are processed by MQSeries link and then either posted to a holding database or switched back into the original application's database. The user is then notified that his information has been returned.

The heart of MQSeries link is a Notes Database named MQSeries link Application Transaction Map (MATM). This is a normal Notes database that is managed and accessed as any other Notes application. This database controls what MQSeries link does as a Notes server task. The MATM provides the following services:

- It manages the fields it receives from the Notes application.
- It handles all security for access to the transaction systems as well as security as the result of those transactions
- It handles all error conditions that result from the interaction with the transaction system.
- It switches the Notes data to a transaction system or switches the transaction back to Notes.
- It handles all interaction with the transaction system.
- It notifies the Notes application as to completion of its task
- It controls the update to any Notes database.

6 Performance

The MQSeries API includes transactional features that enable it to be used in high-performance, time-critical applications. The request handling of MQSeries and MQSeries link was designed to optimize performance. When the information is received by MQSeries link from a Notes application, that Notes application is released to continue on with other functions. It does not need to wait for a reply. In addition, after issuing the request, MQSeries link is also free to handle other incoming requests. It does not wait for that specific reply.