

# The Lotus Notes™ Storage System

*Kenneth Moore*

Iris Associates

1 Technology Park, Westford, MA 01886

moore@iris.com



## Introduction

Lotus Notes is a commercial product that empowers individuals and organizations to collaborate and share information [1].

Notes enables the easy development of applications such as messaging, document management, workflow, and asynchronous conferencing. Notes applications can be deployed globally, across independent organizations, among a heterogeneous network of loosely coupled computers that range in size from small notebooks to large multi-processor systems.

The third major release of Lotus Notes occurred in May 1993. Notes is a client-server product, with clients available on Windows, OS/2, Macintosh, SCO UNIX, HP-UX, AIX, and Solaris. The server is available on Windows, OS/2, Windows NT (for Intel processors), NetWare, SCO UNIX, HP-UX, AIX, and Solaris.

## Storage System

### *Application and information structure*

A Notes application consists of the declarative and functional application description, called the *design*, and the data for the application. A single Notes database contains the entire application's design, but the application's data can be stored in multiple databases.

The Notes data model is unique. It provides intuitive structuring concepts, such as *documents*, *forms*, and *fields*. Documents may contain static information, such as a range of dates, dynamic information, such as an OLE link to a spreadsheet, or actions that are invoked with buttons. Actions are defined with a rich, non-procedural function language, which has syntax similar to popular spreadsheet macro languages. Documents are created and displayed through forms. Forms contain fields, objects, tables, graphics, static text, and buttons.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association of Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.  
SIGMOD '95, San Jose, CA USA  
© 1995 ACM 0-89791-731-6/95/0005..\$3.50

Documents may be organized with views, which are conceptually similar to SQL views, but Notes views have additional reporting features and fewer query capabilities. Doclinks, which are hyperlinks, may be used to associate documents with documents in other databases.

Other components of the database design are the definitions of shared fields (a field definition used in many forms) and macros (procedural actions or asynchronous agents). Also included in the design are help documents for users and administrators of the database, and an icon to identify the database. Databases that only contain design elements, but no data, are called *template databases*. A template database can provide consistent definitions across many database instances, performing a role similar to that of a data dictionary.

The documents and views presented to the user are represented in the Notes storage system as *notes* and *collections*. A note consists of an arbitrary number of items that contain or refer to the data and metadata for the document. There are two kinds of items: *summary* and *non-summary*. Summary items are simple, short data types that comprise the abstract or descriptive information for a document. Non-summary items, which are optional, are for large or complex data types, such as the body field in a mail message.

### *Distribution model*

Most client-server systems assume the client and server are usually connected and any disconnections are for a short amount of time. Conversely, the Notes distribution model assumes that clients and server are frequently disconnected and may only be connected for short periods of time. Many design decisions in Notes were driven by this basic assumption. The Notes model easily accommodates the work behavior of mobile users, as well as conforming to the needs of frequently or continuously connected users.

*Replication* [2] ensures the consistency among widely distributed instances of a database. *Selective replication* provides consistency for a database subset. Unlike log-based or centralized replication schemes, the Notes replication process requires no master list of replicas or global knowledge of nodes containing replicas. Replication is performed on a periodic basis between nodes, with a determination, during the temporary connection, of which

changes to move between common replicas. Any Notes database can be replicated, including template and administration databases.

A Notes network does not require a single point of administrative control. Administration databases, such as the Name and Address Book or Statistics Reporting, provide authorized remote administrators the facilities to control a Notes network. A SNMP-based administration tool, called NotesView, can also be used.

Notes provides discretionary controls and cryptographic protocols to ensure the security of Notes applications in untrusted environments. Secure communication between individuals and controlled information disclosure based on user identity or role within a workflow application are examples of security problems Notes can solve.

### ***Indexing and searching***

Notes provides methods for indexing and searching both summary and non-summary information. Interfaces for querying a database are provided through APIs and the Notes client program. Summary indexing is performed within a collection and presented through a view. Both summary and non-summary data can be searched with full text indexes.

Databases can be efficiently searched for all notes that are new, deleted, or have been changed after a given time. Results of such searches can be additionally filtered to return only those notes that meet content-based selection criteria.

For each database user, Notes maintains in the database a list of unseen or changed documents. Within the desktop database, which is a special database used by the Notes client program, another list is stored. The Notes client and server will synchronize both unread lists so that users who access a database from multiple client computers (for example, office and portable) will see a uniform unread list. Methods to quickly find notes based on their unread state are provided.

A note has a variety of identifiers that are unique within different contexts, for example, a *noteid* uniquely identifies a note within a database instance and an *UNID* uniquely identifies a note across all databases. Notes provides fast methods to access notes by these identifiers.

## **References**

- [1] Lotus Notes Release 3 Documentation Set. Lotus Development Corporation, 1993.
- [2] Leonard Kawell, Jr., Steven Beckhardt, Timothy Halvorsen, Raymond Ozzie, Irene Greif. "Replicated

Document Management in a Group Communication System." In *Groupware: Software for Computer-Supported Cooperative Work*, pages 226-235, IEEE Computer Society Press, 1992.