E-Mail Data Stores

Catherine Horiuchi
Seattle University, USA

INTRODUCTION

For many people, e-mail has become a running record of their business and personal lives. Somewhere in that big clot of e-mail messages that have accumulated over the years is a wealth of information about people they’ve met, work they’ve done, meetings they’ve held. There are tough calls and tender moments, great debates and funny episodes. When did you first meet a certain person? Just what was the initial offer in a business deal? What was that joke somebody sent you in 1998? The answers lie in your old e-mail. Unfortunately, it’s often easier to search the vast reaches of the World Wide Web than to quickly and accurately search your own stored e-mail. (Mossberg, 2004)

Electronic mail, or e-mail, has evolved from its beginnings as one of the earliest Internet applications. The network originally connected computers to computers, but in 1977, RFC 733 updated the messaging protocol to “focus on people and not mailboxes as recipients” (Crocker, Vittal, Pogran, & Henderson, 1977, p.1). Once considered a simple method to send text messages between two machines, e-mail has become a complex system of hardware and software interfaces between individuals and institutions. Messaging technologies manage the flow of major lines of business and significant public sector policy processes. As a corollary to this, databases associated with e-mail now rank among the most mission-critical data stores in many organizations.

BACKGROUND

User-oriented client software interfaces create flexibility to map messages in patterns strongly congruent with the way individuals think and organize information. This usability has resulted in e-mail becoming the catch basin of an organization’s intellectual capital and institutional history, particularly knowledge and activities that are not captured by software systems focused on basic business processes, such as inventory and accounts receivable. The message store grows organically over time. The central message store is linear in nature, with messages stored chronologically, but copies of those messages also are managed by each message originator and recipient. Multiple instances of a message are organized in individualistic fashion in multiple locations, based on each user’s business and personal priorities. Options also exist to create, send, and store messages in encrypted format. Firms face critical decisions regarding e-mail administration including policies on retention, software package management, and mitigation of risks from worms, viruses, and e-mail bombs. Administering e-mail is further complicated by the multiple parties who have the option to discard, retain, or forward an e-mail message, creating yet more copies: the sender, the recipient, the administrator of the originating mail system, and the administrator of the receiving mail system. Table 1 describes the basic location of e-mail messages.

The highly congruent, highly personal aspects of e-mail have contributed to efforts to capitalize on these attributes in an organized fashion. These efforts have varied in approach, and each faces specific challenges, discussed in the following section.

DATA MANAGEMENT CHALLENGES

Strategies to capture the knowledge held in e-mails have ranged from benign neglect (e.g., limited to backing up and archiving a central e-mail message store), to direct integration with a primary business application (e.g., using mail message protocols within a supply-chain software package, such as SAP), to sophisticated programming to capitalize on a particular e-mail platform (e.g., business application programming on Lotus Notes). Each approach is complicated by authentication, platform dependence, data corruptibility, and referential integrity issues.

The simplest strategy, benign neglect, is also the most common: The message store is backed up with the rest of the data on the server. If a user inadvertently deletes a message considered important, a request to the system administrator can result in it being restored from backup. This strategy can also meet legal requirements to retain e-mail if the system administrator has been notified of the requirement and has adequate storage capacity and processes in place. However, it is also dependent on the original sender/recipient to reestablish the connection between the particular e-mail message and its context among many issues and correspondents. And if the mes-
E-Mail Data Stores

Table 1. Where e-mail data is stored

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>On individual, end-user systems</td>
<td>A common approach for POP3 users. Messages are copied from a mail server message store onto an individual’s computer. The messages are then deleted from the server.</td>
</tr>
<tr>
<td>On servers</td>
<td>A common approach for IMAP users with web-based mail clients. Messages are stored only on the mail server.</td>
</tr>
</tbody>
</table>

Most mail systems are configured to store messages both on users’ machines and in a central database repository. Large message stores are usually included in system backups, resulting in further replication of messages and attachments.

sage was encrypted, loss of the original key will result in a need for a brute-force decryption, a time-consuming process. This simplest strategy also fails to address referential integrity, the principle that assures all copies of data are congruent and no part of the data loses its association with other elements. For instance, if an attachment is deleted, the message is incomplete; if an Internet site linked to a message is altered or expires, the message is deleted, the message is incomplete; if an Internet attachment with other elements. For instance, if an attachment is deleted, the message is incomplete; if an Internet site linked to a message is altered or expires, the message no longer retains the same meaning.

Organizations with multiple hardware and software systems struggle to collect and analyze data at the organizational level (i.e., metadata analysis). To improve connections and reduce the time required for metadata analysis, firms may replace numerous free-standing applications with a corporate resource management (ERP) package, or combine data sources into a more integrated data warehouse (Zeng, Chang, & Yen, 2003). These ERP and data warehouse solutions include hooks for messaging technologies to establish links between departments and even external companies in supply-chain automation. This linking technology is compatible with several major e-mail engines so that firms can leverage their existing e-mail systems’ knowledge in service to these specialized applications.

Combining a corporate-level e-mail system with a corporate-level business software package automates many manual processes and creates a strong audit trail. However, this creates a high degree of dependence on the technology companies who license the software products as well as any consultants who may have been hired to write specialized software routines and database extensions targeting particular business process automation. These type of technology enhancement projects easily run into the tens of millions for initial implementation and millions for annual maintenance. ERP packages and similar integrated software strategies address the referential integrity problem inherent in having multiple systems that describe various aspects of a single transaction. Instead of separate systems that catalog the purchase of a piece of equipment—its installation at a location, maintenance schedule, depreciation, ultimate removal, and salvaging—a single system tags all these events to the equipment, resulting in a more comprehensive picture. Although this operational cohesion is of high value to management, a firm’s dependence on particular vendors results in loss of competitive pressure. Transitioning to an alternate vendor involves major expense, as does changing the integrated system to meet new business requirements. Historically, firms used software/hardware packages for decades, but that was before software tightly programmed employee behaviors that must shift with changing economic cycles and market challenges.

Authentication between major applications and external data stores can be handled in more than one way, with differing security profiles. The least satisfactory method, from a database administrator point of view, assigns administrative privileges to an application. The database administrator does not control rights of users in the application and cannot match data requests to users. It also exposes the data store to hacking exploits designed to enter the data store using other methods, such as a query engine inherent to the database management system. An alternative method, with named users and permissions at both the database and application level, may require users to authenticate multiple times. This basic dilemma underlies efforts for “single sign on.” In most instances, an end user has several separate entities established with the operating systems of multiple servers as well as applications, and a portion of their authentication has been established in a pass-through or permissions-table fashion. A firm risks security compromises to information based on the degree to which it maintains an active user directory.

Rather than buying a software package and using the messaging software merely to route transactions, other firms have centered on the messaging itself and written extensive software enhancements to the basic message function. This is the Lotus Notes strategy. It is best suited for firms with substantial intellectual property, as opposed to firms with extensive inventory to track or manu-