DELEGATING AUTHORITY IN A DISTRIBUTED INFORMATION MANAGEMENT SYSTEM

Kareem S. Aggour, Barbara J. Vivier, Janet A. Barnett
GE Global Research, One Research Circle, Niskayuna, NY, 12309, USA

Keywords: distributed administration, delegate authority, LDAP, information management, community management, identity management, Single Sign On

Abstract: The need to manage large information repositories in a secure, distributed environment increases with the growth of the Internet. To address this need, a system capable of managing the contents of an LDAP directory over the Web has been designed and developed. This system allows for the directory's data to be divided into communities and supports the delegation of administrative authority over those communities to a distributed set of administrators. The communities may be subdivided recursively into subgroups, and rights over those subgroups also may be restricted. Thus, system administrators can dynamically delegate subsets of their permissions over a subset of their managed data, allowing for the flexible and effective control of permissions over the data within distributed organizations. The system solves the delegated administration problem for managing the contents of an LDAP directory in a distributed environment. Today, it supports the administration of over 20 production directories by well over 2000 distributed administrators.

1 INTRODUCTION

The objective of this research was to design and develop a distributed information management system capable of remotely managing the information in a Lightweight Directory Access Protocol (LDAP) database, utilizing a flexible model of delegation and permission control. LDAP is a hierarchical database optimized to execute fast data reads, even across extremely large volumes of data (Weltman & Dahbura, 2000). One of the key capabilities of the system is that it facilitates the division of the LDAP data into logical groups (communities).

The system also allows for the delegation of administrative authority over those communities to the experts most capable of managing each portion of the data, regardless of their location. The solution is responsible not only for giving users access to the data they need to administer, but also for preventing users from viewing or modifying information they are not authorized to view or edit.

Another key capability is that the system operates independently of the underlying LDAP schema (data structure) and, therefore, is capable of managing any LDAP directory. Similarly, the system functions without having to modify the data structure of the managed directory in any way. Finally, as it operates in a distributed environment, it operates over the Internet utilizing only standard Internet protocols such as Hyper Text Transfer Protocol (HTTP).

Today, the system allows for millions of records distributed across multiple disparate directories to be administered by thousands of globally distributed administrators.

This paper describes the design and implementation of the Community Management Tool (COMET), a distributed information management system. Section 2 provides a background explaining the need for COMET and describes prior art. Section 3 describes the COMET approach for creating communities and for delegating authority to users. Section 4 provides some implementation details. Section 5 gives some results of this effort; Section 6 presents future opportunities; conclusions are given in Section 7.

S. Aggour K., J. Vivier B. and A. Barnett J. (2004).
DELEGATING AUTHORITY IN A DISTRIBUTED INFORMATION MANAGEMENT SYSTEM.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 155-162
DOI: 10.5220/0002602801550162

2 BACKGROUND

Many businesses have adopted the Web as a vehicle for delivering both information and services to customers, suppliers, and employees. By moving internal business processes and external service offerings to the Web, businesses can achieve operating efficiencies and cost reductions. Businesses that efficiently provide applications to customers or suppliers via the Web can increase the number of customers they serve without appreciable increases in operating costs.

As the number of such applications grows, it is desirable to both the business and to its application users to provide a single login to each user. For the user, access to multiple applications is simplified. For the business, access to applications can be controlled and monitored more readily.

General Electric (GE) has digitized many of its internal processes for its hundreds of thousands of employees worldwide. Most of GE's businesses also provide Web-based products and services to their global customers and suppliers. Because of the size of its user community and the large number of applications, GE established a *Single Sign On* (SSO) initiative across the company (Loshin, 2001).

The goal of SSO is to have all Web-based applications share the same data repository for user IDs, passwords, and other common information. SSO benefits GE in that application programmers no longer have to worry about collecting and managing the common information. The cost and time to develop Web applications is reduced, as is the cost of maintenance and help desk support. GE's SSO solution, while achieving the intended benefits, had its drawbacks, however. Millions of global employee and customer records are located in a single repository. Therefore, shared administration is necessary to manage the information effectively. Compounding the problem, the administrators who are most capable of managing the data are as distributed as the end users. Thus, a distributed information management system is required to distribute authority to a global community of administrators responsible for managing this huge volume of information.

At the same time as the SSO initiative, some of GE's businesses began to offer Web-based services to *communities* of customers. For example, a GE business may contract with a customer to provide a suite of Web services for the customer's staff. The management of a user community requires capabilities in addition to those for Single Sign On. A community-based service compounds the challenge of managing the user directory because the knowledge of the users in the community resides within the community, rather than at the GE business. Therefore, GE must provide a mechanism to allow communities of users to be established within GE's repository but maintained externally.

2.1 Prior Art

While the use of LDAP is growing, the number and sophistication of LDAP administration tools have not grown at the same rate. Two commercial tools were found in an attempt to address this need; however, each provided only a partial solution to the problem.

Oblix's Secure User Management Solution (now a part of Oblix's NetPoint product) (Oblix NetPoint, 2003) is capable of delegating the administration of subsets of data and also supports specifying attribute-level permissions on the data for administrators. (An LDAP entry is comprised of a set of 'object classes' that have corresponding 'attributes.' An entry has any number of these associated attributes, which may be single or multi-valued (Weltman & Dahbura, 2000).) However, Oblix does not support arbitrary levels of delegation, i.e., administrators cannot subdivide their world and give other users part of their administrative authority. Oblix also does not support dynamic assignment of users to groups. Oblix's group model assumes that the organization is using LDAP groups to arrange its user communities, an assumption that restricts the structure of the customer's directory. LDAP groups are objects comprised of a list of members.

At the time, Netegrity's Delegated Management Services (DMS) system was in the early phases of being released as version 1.0. Offering less capability than the Oblix solution, it did not support a sophisticated model of attribute-based authorization, supported only one level of delegation, and enforced restrictions on the LDAP group structure. Companies with an existing LDAP infrastructure would have difficulty using Netegrity's DMS system. Netegrity has since released the product IdentityMinder to replace DMS (Netegrity IdentityMinder, 2003). IdentityMinder supports role-based access control, although it still lacks the flexibility GE requires.

While there are now more vendors in the emerging area of *identity management,* these vendors focus on access control and Single Sign On to enterprise applications rather than on the challenge of distributing administration of a large directory (Senf, 2003). Identity management does not provide a flexible delegation model to support multiple overlapping or nested or isolated communities of users. Identity management across a

large set of applications also requires, but does not typically provide, administrative access control at the attribute level. To date, no commercial product satisfies all of the requirements for distributed directory administration met by COMET.

3 SOLUTION DESIGN

COMET is a Web-based delegated administration information management system used to maintain the data within an LDAP directory in a distributed environment. Two concepts are key: *Domains* and *Authority*. COMET allows LDAP data to be divided into groups and subgroups. Specific administrative permissions over those groups may then be defined and limited. This combination of a logical subset of data combined with permissions over that data is referred to as a *domain*.

User accounts within the LDAP directory may be assigned administrative *authority* over the domains. Administrative authorities come in two flavors: *Delegate* and *Edit*. An administrator with *delegate* authority may divide an existing domain into sub-domains and may assign (delegate) authority over those sub-domains to other users. An administrator with *edit* authority may edit user information within a given domain, subject to the domain's permissions. COMET also allows users to edit their own account information (within configurable limits). A complete delegated administration information management system, COMET enables the creation and management of groups (communities) of LDAP data.

COMET was designed to function with any LDAP directory schema. Everything that COMET knows about a directory is discovered programmatically. COMET can determine what object classes are defined in a directory, what attributes are defined for those object classes, and whether each attribute is single or multi-valued. This flexibility allows users to specify the directory information to be managed. The user can select the object classes and the attributes of interest. If the attribute is multi-valued, the user can also opt to make the attribute appear single-valued to COMET. This ability to discover the underlying schema of an LDAP directory is critical to COMET's functioning; it allows COMET to manage arbitrary LDAP directories.

COMET was also designed with the ability to manage multiple disparate directories with a single software installation. This functionality allows customers to have a common graphical user interface and a single point of executable code to manage very different directories. For example, companies can manage their internal employee information, external customer data, and product catalog information all with one COMET installation. The term *configuration* refers to that subset of a particular LDAP directory managed by COMET. A COMET configuration contains all of the users in a managed directory and identifies an end-user's view, edit, and delete permissions over his or her own data; whereas a COMET domain defines an administrator's view, edit, and delete permissions over the data for users that fall within the domain.

3.1 Domains

Domains are used to divide the directory into manageable groups and subgroups. The properties of a domain are described in Table 1.

Table 1: Domain Attributes

Required Attributes	
DomainID	Unique ID of the domain
DomainName	Display name of the domain
QueryRule	Defines the domain user community
ParentDomainDN	DN of the parent domain
Optional Attributes	
AttrDeleteable	List of deleteable attributes
AttrEditable	List of editable attributes
AttrViewable	List of viewable attributes
DomainDescription	Text description of the domain

In the domain hierarchy, there is a single root domain for each configuration called the Super Admin Domain. All subsequent domains are descendants of this domain. The Super Admin Domain contains all users in the configuration. This domain can be divided into sub-domains that have fewer (or the same number of) users and fewer (or equal) administrative permissions. No domain can contain a user or administrative permission not found within its parent domain. However, two separate domains can have overlapping groups of users associated with them, as well as overlapping permissions. Figure 1 shows two example Venn diagrams of possible user group structures; each oval represents a community.

Since domains are hierarchical, deleting a domain first results in the recursive deletion of all of its child domains and the removal of all administrative permissions over the domain. Any administrator records that reference the domain are removed.

3.1.1 Community Definition

An LDAP search pattern identifies the users within a domain; the pattern returns all users who satisfy the search criteria. In COMET this pattern is referred to as a *Query Rule*. The example shown in Figure 2 can be read as: "Select all users whose Company Name is 'GE' or 'General Electric'."

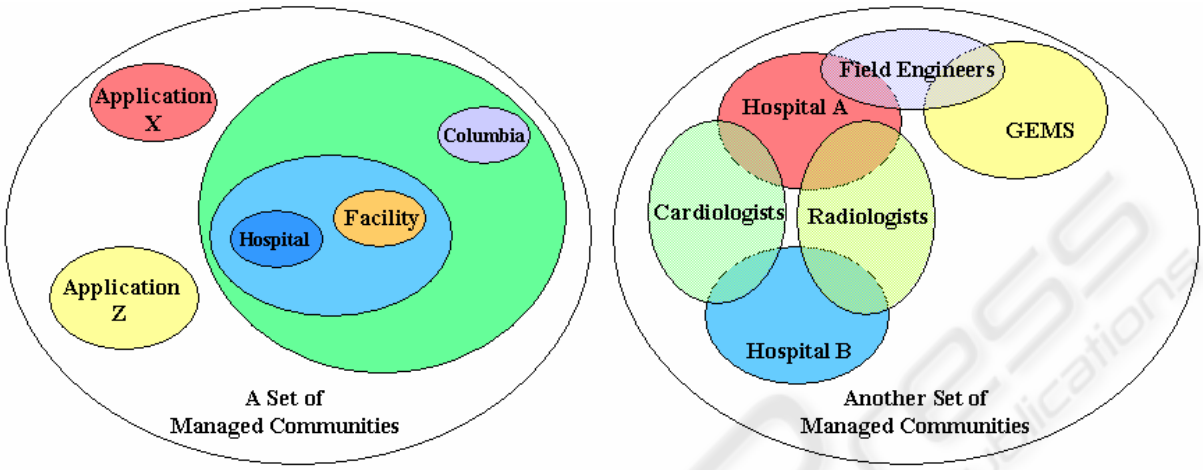


Figure 1: Sample Domain Venn Diagrams

COMET query rules are in prefix notation, consistent with standard LDAP notation (Weltman & Dahbura, 2000). Also in keeping with LDAP notation, query rules utilize the character '|' to represent an OR conjunction and the character '&' to represent an AND conjunction.

By using query rules to establish domain membership as opposed to listing individual users (as is the case with standard LDAP groups), COMET allows domain memberships to change dynamically as the data in the directory change. As employees join and leave the General Electric Company, their Company Name will change and they will automatically be added to or removed from the 'GE Employee Domain' defined by the query rule in Figure 2. This group definition model requires the least effort for administrators to describe and control the membership of a user community.

```
( | (companyname=GE)
(companyname=General Electric) )
```

Figure 2: Example COMET Query Rule for a 'GE Employee Domain'

When an administrator executes a search for users within a domain, it is important to ensure that the users returned from the query are constrained by the hierarchy of domains. Therefore, the administrator's search rule is concatenated with a query rule formed by recursively concatenating the query rule of the current domain with that of each of its ancestor domains, up to the Super Admin Domain. This recursive concatenation, using an AND operator, guarantees that no user will be found in a sub-domain without being in the parent domain.

Query rules can be created in two ways in COMET. A Query Rule Wizard tool supports the definition of a query consisting of no more than six rules joined using the conjunctions AND and OR. The query rule wizard concatenates the individual rules such that the final query is of the form (x(x(x12)3)4) where '1' through '4' are the rules and 'x' are the conjunctions. Each rule is defined in the form: '(attribute operator value)'. The 'operator' is a comparison operator such as '=' (equals), '!=' (not equals), etc. The 'value' supports pattern matching and may include '*' for wildcard matching. A screen capture of the Query Rule Wizard user interface is shown in Figure 3.

COMET also provides an interface for entering custom query rules that must be standard LDAP queries but can be of arbitrary form.

3.1.2 Permission Definition

A domain contains sets of permissions describing an administrator's rights to view, edit and delete user attribute values. Attributes that are available to be assigned to a domain are only those available to that domain's parent domain. This prevents assigning rights to a child domain that are not available to the parent. The Super Admin Domain has no 'parent

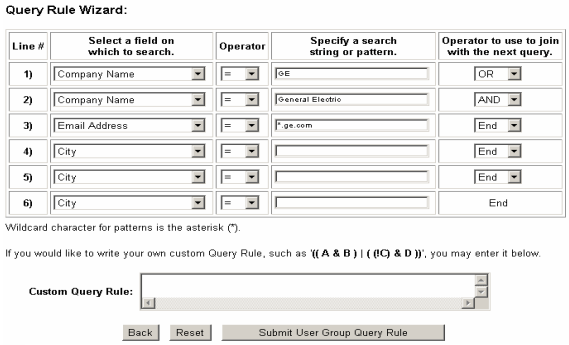


Figure 3: Query Rule Wizard User Interface

domain' so the list of editable attributes is obtained from the list of all managed attributes in the configuration. All view, edit, and delete attribute permissions are available to the Super Admin Domain by default.

3.2 Administrators

The administrator attributes are shown in Table 2. An administrator record in COMET contains the DN of the user who is an administrator, the DN(s) of the domain(s) over which they have edit authority and, separately, the DN(s) of the domain(s) over which they have delegate authority.

There are several types of administrative capabilities within COMET. The *Master Domain Administrator* (MD Admin) is the UNIX-style 'root' user of COMET. The MD Admin has complete authority over COMET across all configurations and is the only administrative account with the right to create and delete configurations (i.e., add and remove references to managed LDAPs). It is a separate account defined during installation, whereas all other administrative permissions are delegated to users within the managed LDAPs. A *Configuration Administrator* has the same root-like power as the MD Admin but limited to a single configuration. Only the MD Admin and Configuration Admin can adjust the LDAP fields that can be managed within an entire configuration.

The *Super Administrator* has authority to manage the Super Admin Domain and any sub-domains within the configuration. For this reason, the Super Admin is distinguished from a regular administrator.

An administrator with *Delegate Authority* has the ability to divide an existing domain into sub-domains, as well as the ability to delegate administrative authority over those sub-domains to other users.

Edit Authority permits an administrator to edit user information in a given domain, within the bounds defined by the domain constraints. Edit authority may be granted by an administrator with delegate authority over the domain.

End users have no assigned authority but may view and edit their own account information, within limits set in the configuration. It is possible to define a configuration such that end users may be denied the ability to view or edit any of their own information. Users can view their most direct administrators, i.e., the administrators with edit authority over the lowest-level domains within which the user belongs. This allows users to contact their administrators and request changes to their accounts.

Table 2: Administrator Attributes

Required Attributes	
AdminID	Unique ID of the administrator entry
UserDN	DN of the administrator
Optional Attributes	
EditAuthority	List of domain DNs with expiration timestamps
DelegateAuthority	List of domain DNs with expiration timestamps

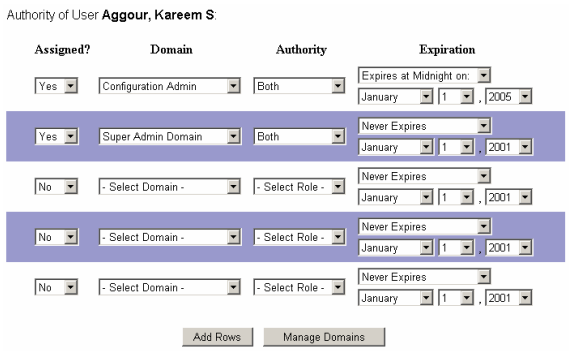
3.3 Delegation of Authority

Delegating administrative authority is the mechanism for establishing which administrators can manage which communities of users. Every delegated administrator has authority over at least one domain. An administrator can be granted delegate authority, edit authority, or both. A user can be an administrator for several different domains, and also can have different authorities in those different domains.

A typical delegate administrator can grant authority only over the sub-domains of the domain over which they have delegate authority. Alternatively, an MD Admin, Configuration Admin, or Super Admin is able to delegate authority over any domain in the configuration. The MD Admin and Configuration Admin can also delegate Configuration Admin capabilities to users. When a user's existing authorities are displayed, as seen in Figure 4, the delegate administrator can view only those authorities he or she has the ability to grant or revoke.

As seen in Figure 4, four columns are displayed on the COMET delegate authority user interface. The first column indicates that the row is active. Switching a 'Yes' to a 'No' and submitting the form will revoke the user's authority over the specified domain. The next column is a drop-down of the domains that can be delegated, only one of which may be selected in each row. The third column

contains the type of authority to be delegated. Three options are available: 'Edit', 'Delegate', and 'Both'. The 'Both' option simply grants both authorities at once. The final column allows the administrator to specify whether or not the authority being granted is permanent or temporary ('Never Expires' or 'Expires at Midnight On:'). If the expires option is



selected, then a month, day, and year must be selected from the corresponding drop-downs. A user may have multiple authorities, but each has its own expiration date (and some may not expire).

Figure 4: Delegate Authority User Interface

4 SYSTEM ARCHITECTURE & IMPLEMENTATION

A three-tiered, component-based architecture using object-oriented design paradigms was used to manage a large volume of administrators and directories in COMET. COMET is written entirely in Java 1.2. JavaServer Pages (JSPs) handle the presentation layer. Servlets and Java classes comprise the application/business logic layer. Finally, Java classes, LDAP directories, properties files, and text logs act as the data layer.

4.1 Data Storage

All COMET information is stored separately from the LDAP data being managed. For example, assigning a user an administrative authority will not modify that user's account information. Figure 5 shows a clear separation between the "COMET LDAP Directory" and the "Managed Directories." COMET stores most of its information in a COMET-specific branch of an LDAP directory. Therefore, the COMET data can be stored within the same physical directory as the data being managed or it may be stored in a completely separate LDAP instance; either way, COMET information will not intermingle with the managed information. This allows for the easy installation and removal of COMET. By not requiring any COMET information to be stored with the managed data, the managed data structure does not have to be modified in order to be managed by COMET.

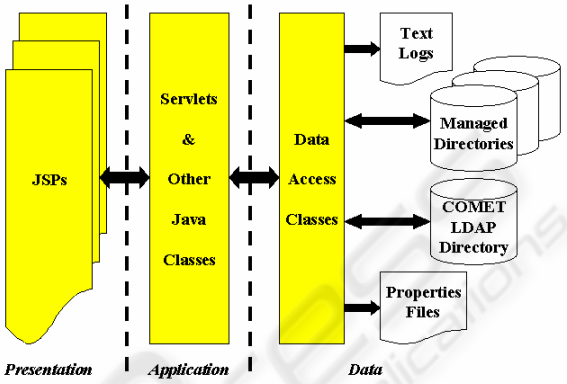


Figure 5: Three-Tiered Architecture

There are also a number of properties files in the data layer. The critical property file contains the information that allows COMET to connect to the COMET LDAP directory, including the host name, port, user ID, and password. This file also contains the user ID and password of the MD Admin account stored as a one-way hash so they cannot be recovered. All other security-related information, such as user IDs and passwords used to connect to the managed LDAP directories are encrypted and stored directly in the COMET directory.

LDAP is a hierarchical data structure, so all objects are placed at specific locations in the directory 'tree' (Weltman & Dahbura, 2000). Almost all system information is stored in the COMET LDAP directory beneath the special COMET branch. There is a branch for each configuration within the hierarchy as seen in Figure 6. The root of each configuration branch contains the configuration information. Beneath each configuration branch are two branches, one each for domains and administrators. Additional branches support functionality not described in this paper.

4.2 Additional Functionality

COMET is a complete, secure, distributed information management system; it provides a whole host of functionality. Additional functionality includes a security mechanism responsible for user authentication and authorization, session

management, LDAP group management, and logging.

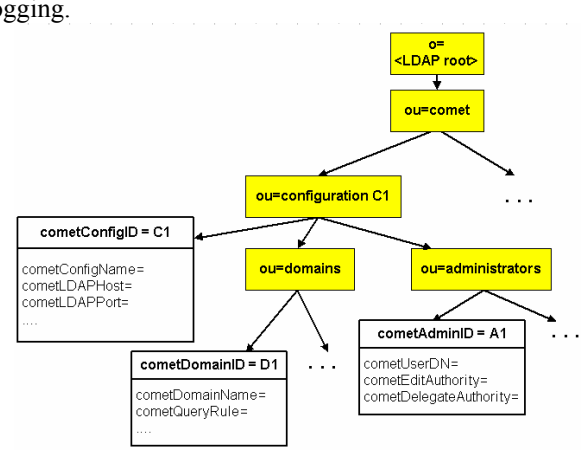


Figure 6: LDAP Tree Structure for Configurations

For authentication, COMET features a login page to which unauthenticated users will be automatically redirected when they attempt to access any secure COMET page. A session class maintains data including what configurations the user may access and if the user is an administrator in the current configuration. A temporary browser cookie maps each user to his or her COMET session. Static LDAP group management is also supported. COMET enables the addition and removal of users to and from LDAP groups, including a sophisticated method to identify and select which users to add or remove from a particular group.

COMET logs messages (from inconsequential 'debug' to system 'critical' messages) in a text log file, and also logs all changes to each LDAP directory. The LDAP change logging stores the previous and new values whenever an object is changed through COMET. This enables system administrators to track who makes what changes.

This is just a small taste of the additional functionality available in COMET, existing in order to ensure COMET's practical applicability to the distributed information management problem.

5 RESULTS

COMET has been running in production for over two years and is used extensively across GE for managing LDAP directories containing both internal employee and external customer records. The ability to manage directories of arbitrary structure by dynamically learning about the underlying schema has been critical to its broad acceptance within GE. Another key component is its ability to manage a directory without having to modify the directory's data structure. These features allow each GE business to maintain a unique schema while using the same COMET installation.

The GE businesses' Help Desk organizations are the principal users of COMET as they field support calls. COMET is the first tool used by the staff when they answer an internal support call, and has thus become central to their work. COMET's ability to assign administrator permissions at the attribute level within a domain enabled the creation of Help Desk Domains with limited access to view and edit attributes.

Several GE businesses are also using COMET to manage communities of their customers. The ability to define communities of users and to separate edit authority and delegate authority over those communities is critical to these GE businesses, as they delegate and distribute limited authority to different internal and external organizations. Because of these benefits, there are now well over 2,000 administrators managing hundreds of thousands of user records for both internal employees and external customers and suppliers. These administrators use COMET extensively. COMET averages about 600 logins per day and can reach over 900 logins per day.

COMET has been installed in production at several locations within GE. Together, these installations are currently managing over 20 different LDAP directories, demonstrating COMET's ability to manage multiple, disparate directories with a single installation.

One of the primary benefits of COMET is its flexible approach to the LDAP schema. Since COMET makes no assumption about the schema, any LDAP data structure can be managed. However, this flexibility requires that the user have a fairly solid understanding of LDAP in order to configure the system most efficiently. The configuration manager must be knowledgeable about the underlying information repository.

Unlike many identity management packages, COMET cannot assign administrators based on their attributes, i.e., groups of users cannot be assigned authorities. In COMET, authorities are delegated to specific individuals. While maintaining strong control over authority, this approach is less flexible than other role-based approaches.

COMET is most beneficial when the community of administrators is highly distributed and when flexibility is needed regarding who can view and/or edit what attributes. If there is a single administrator or a small set of administrators who know LDAP, who are centrally located, and who have the same privileges over the data, then COMET may not be necessary. COMET would still provide an effective

graphical user interface to view the data, but it may not provide a measurable benefit.

6 FUTURE OPPORTUNITIES

COMET could be extended in several directions, most especially for: manipulating LDAP schema, incorporating LDAP static groups, and addressing non-LDAP data. COMET currently manages the data within a directory. COMET could be expanded to be an LDAP data *definition* tool to manipulate object classes and attributes within the underlying directory schema. Current methods for Sun ONE Directory Server (Sun ONE Directory Server, 2003) schema manipulation involve either modifying the LDAP properties files by hand or using Sun ONE's LDAP Management Console graphical user interface, which we have found to be cumbersome.

COMET currently identifies user groups through the use of query rules. LDAP supports the notion of static groups to which users are individually added. This enables groups of users to be formed that have no common attribute values. It would be useful if COMET could also use LDAP groups to identify the user community within a domain. COMET already supports managing LDAP groups, so this would be a natural next step.

COMET's architecture is independent of the underlying directory structure. COMET has been developed as an LDAP information management system; however, it could be expanded to manage databases other than LDAP directories. If COMET could also manage relational databases, it could become a general delegated administration information management system capable of managing many types of repositories.

7 CONCLUSIONS

COMET enables GE's businesses to manage customer and employee data globally and effectively by allowing the delegation of authority to local administrators at customer sites, thereby enabling those customers to manage a subset of their own data.

A growing number of GE businesses are using COMET for GE's Single Sign On initiative. Despite the many disparate directories being managed, no business has required any customizations to COMET. COMET is simultaneously delegating the management of multiple, disparate directories in a distributed environment.

COMET has enabled GE Help Desk personnel to better handle support calls, increasing their efficiency while reducing maintenance costs. Similarly, GE businesses are able to improve their customer's online experiences by offering functionality to communities of users without burdening GE to manage all of the records in each community. Through COMET, the most capable administrators are able to manage portions of the data regardless of their location.

ACKNOWLEDGEMENTS

We appreciate the support of our management, especially Andy Deitsch, and the technical collaboration of Mark Kornfein, David Mehring, Osman Öksoy, and Wayne Uejio.

REFERENCES

Loshin, P. (2001). *Single Sign-on*. Retrieved January 19, 2004, from, http://www.computerworld.com/security topics/security/story/0,10801,57285,00.html

Netegrity IdentityMinder: Overview. (n.d.). Retrieved October 9, 2003, from, http://www.netegrity.com/ products/products.cfm?page=IMoverview

Oblix NetPoint. (n.d.). Retrieved October 9, 2003, from, http://www.oblix.com/products/netpoint/index.html

Senf, D. (2003). *Identity Management: Securing Your E-Business Future*. Retrieved January 19, 2004, from, http://www2.cio.com/ analyst/report940.html

Sun ONE Directory Server 5.2. (n.d.). Retrieved October 9, 2003, from, http://wwws.sun.com/software/ products/directory_srvr/home_directory.html

Weltman, R. & Dahbura, T. (2000). *LDAP Programming with Java*, Addison-Wesley, Reading, MA