# An Architecture to Support Communities of Interest Using Directory Services Capabilities

David Kuechler
*Georgia State University*
*dkuechler@acm.org*

Vijay Vaishnavi
*Georgia State University*
*vvaishna@gsu.edu*

Art Vandenberg
*Georgia State University*
*avandenberg@gsu.edu*

## Abstract

*Directory services provide a mechanism to describe internal resources that can be dynamically discovered and used, and are increasingly being adopted by organizations. Despite being in an era of extensive interorganizational collaboration, relatively little attention has been given to extending directory services across organizational boundaries. In particular, resources described using directory services are either (a) constrained by description possibilities or (b) cannot be widely understood due to idiosyncratic description mechanisms. Thus, current directory services approaches are not well suited for the information requirements of groups of individuals spanning organizational boundaries: communities of interest (COIs). A COI is a virtual community that benefits from electronic communication, especially by the ability to provide structured descriptions of relevant resources. This paper describes a new approach to support COIs through directory services by enabling autonomous entities to provide structured, evolutionary descriptions of members and resources while retaining directory services' core strength: structured, semantically coherent descriptions of resources. The requirements architecture is derived, a walkthrough of the architectural elements is given, and a research prototype is briefly described.*

Keywords: directory services, LDAP, communities of interest, semantic heterogeneity

## 1. Introduction

Directory services provide a well-defined and general mechanism for describing resources within an organization and enable their discovery by other individuals and applications [1]. While arguably a simple concept, the appropriate use of directory services is being recognized as a key to competitive advantage [2]. Given the increasing focus on organizational learning and the potential for directory services to consolidate important facets of organizational knowledge this is not surprising; indeed, a core issue in knowledge management is the identification of potential sources of knowledge [3, 4].

While directory services have focused on sharing information, it has primarily focused on doing so within an organization and relatively little attention has been given to sharing information across organizational boundaries. Increasingly, a core source of competitive advantage is recognized to consist of making optimal use of internal resources and potential *external* resources. Compelling arguments have been advanced that the appropriate use of external resources is not only necessary, but also essential to success in the future business environment [5].

We argue that two particular problems arise in the use of directory services between autonomous groups. First, semantic heterogeneity issues arise when attempting to use information from disparately created data sources (e.g., synonym or homonym problems) [6, 7]. In other words, different vocabularies are used in various organizations to describe concepts that may or may not be the same. Second, current directory services are based on a traditional view of data storage: pre-defined schemas are used to describe various resources. These fixed descriptions usually prove to be inadequate as requirements continually evolve over time [8].

While directory services are capable of describing resources in general, this research focuses on the description of resources relevant to one or more communities of interest. We have framed the problem of interorganizational collaboration as being at the community awareness level [9]. We use the term *Community of Interest* (COI) to denote a collection of individuals who share a common interest, and who are supported by access to structured descriptions of individuals and resources relevant to that community, with agreed upon semantics for the structured

descriptions. In particular, this paper focuses on the following research problem:

*How can directory services be extended to support COIs by enabling individuals to provide flexible, complete descriptions of themselves and relevant resources that can be accessed effectively by individuals from different organizations (and organizational units)?*

## 1.1. Motivating example

The above research question is general. As a specific example of a COI, consider the community of individuals who wish to share not only specifications for access to local video equipment, but also knowledge about video usage and video experts in general. This community currently uses the "Commons Portal," which is intended to enable individuals with differing levels of experience (beginners, intermediate, expert) to find relevant videoconferencing resources. A database stores the links, Universal Resource Locators (URLs), and other information that has been assembled over time, and this information is made accessible through a web site. However, there are several significant limitations to this approach:

- The provision and update of the latest available information is constrained to the very few individuals who have access to update the Commons Portal web site (e.g., information about a new video camera may be available, but cannot be provided directly by a user).
- The extension of video camera specifications (i.e. metadata) is constrained to these same individuals (e.g., settings for backward compatibility with older cameras may be known, but cannot be provided in a structured format).
- The description of individuals who may know about or be interested in testing a particular piece of video equipment is similarly impaired since publications to the web site are restricted to the few with direct update access.

These examples are intended to motivate the need for *flexible* structured resource description and discovery within a community. Clearly, a discussion bulletin board could potentially meet each of the information requirements described above, but the representation of any information would be unstructured. Similarly, given diligent administrators of the web site, each of these needs will eventually be met by appropriate modifications to the structured representation of information on the Commons Portal, but these changes will also entail significant use of the administrator's time and publishing delay. This research is intended to facilitate the structured representation of information by individuals

from different organizations, without requiring the delays of central approval.

## 2. Resource description and discovery: current approaches

The ideal solution is indicated by the research question itself: (1) individuals are able to provide structured but flexible descriptions of resources, and (2) other individuals are able to query these descriptions to discover resources relevant to a particular project or interest. This section describes current approaches to resource description and discovery, together with their limitations as compared with this ideal.

The World Wide Web provides some of these capabilities as an extremely flexible description repository that is widely accessible. By publishing a personal web site, individuals can describe any resource using their own choice of terms and media, and these descriptions can be queried by other individuals using various search engines. However, any description on the Web is likely incomplete; since an individual may not realize that other people share a common interest, his or her description of a resource may be limited relative to that interest. More significantly, free form descriptions enable only keyword searches; this limits the precision of searches [10] and the automated processing of information [11].

Directory services help to address these issues by providing a mechanism for the *structured* description of a resource. While there are several approaches for providing directory services, we focus on Lightweight Directory Access Protocol (LDAP) [1] because it is an open standard, widely supported, and clearly defines key directory concepts. A directory consists of a set of entries, each of which corresponds to a particular resource; each directory entry is uniquely identified by a distinguished name, which can be thought of as a primary key in relational database (RDBMS) terminology. Each entry is then further described by a set of attribute-value pairs; e.g., an individual might have an attribute "email," followed by the particular value "johndoe@some.org." Furthermore, each entry is a member of one or more objectclasses, which are analogous to the concept of a schema in an RDBMS. Each objectclass specifies the attributes that *may* be present to describe a member of that class, and those that *must* be present. Unlike a traditional RDBMS, attributes may be multi-valued for a directory entry. Thus, the attribute "email" for a directory entry might have the values "johndone@some.org" and "johndoe@isp.com."

Despite the flexibility offered, there are significant limitations to this approach: only certain *pre-specified* fields can be used in the description of any directory entry. In other words, individuals cannot add new fields on their own volition to describe resources in greater detail. In addition, in an interorganizational setting, the semantics of new fields may be different across directories. The same attribute name may be used to signify entirely different things in different organizations, and different organizations may use different attributes to describe the identical concept [7].

Centrally controlled standards-based approaches largely overcome the semantic heterogeneity issue. However, their limitations can be seen through the evolution of message formats within EDI [12] and within directory services itself as classes available to describe individuals have evolved from *orgperson*, to *inetorgperson*, to *eduperson* in an attempt to fulfill ever expanding descriptive requirements. Beyond this empirical evidence, there are significant theoretical arguments that any fixed descriptive languages will necessarily prove to be incomplete over the long term [8].

One solution to extend directory services is the "directory of directories" approach [13]. This approach replicates particular attributes from underlying source directories into a master directory, and enables queries to be issued against this master directory. While this approach addresses important scalability issues in querying a large number of independent directories efficiently, it does not enable extensible descriptions.

Directory Services Markup Language (DSML) [14] provides a significant advance by enabling the contents of directories to be published as XML documents that can be processed syntactically by XML-aware applications. However, DSML provides only a specification for the translation of LDAP information into XML, and does not enable more flexible descriptions or semantically homogeneous queries.

## 2.1. Flexibility versus homogeneity of descriptions

Figure 1 provides a schematic representation of current information representation and sharing approaches with respect to flexibility and homogeneity of descriptions. The leftmost extreme contains fully unstructured descriptions. HTML-based descriptions are placed in this realm since they provide extremely flexible descriptions, but do not provide any guarantees as to the structure or semantics of the information provided. This relegates any queries to keyword searches.

Just to the right of the "least" structured descriptions are fully flexible "structured" descriptions. Extensible Markup Language (XML) falls in this category. XML is

an extremely flexible mechanism for expressing information, while providing structure and constraints on the data that is expressed [15]. XML provides the advantage of HTML's expressiveness together with the potential for structured queries, but does not provide semantic interoperability (i.e., comprehensibility of attributes and attribute values) among publishers of information and users querying that information.
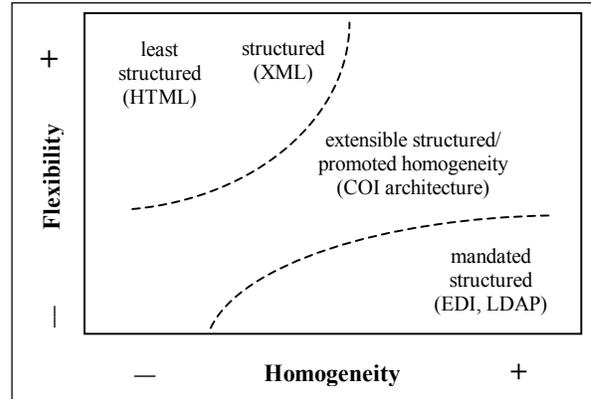


Figure 1. Flexibility versus homogeneity of descriptions

"Mandated structured" descriptions fall at the rightmost extreme. Information expressed through standards such as Electronic Data Interchange (EDI), organizational RDBMSs, or current directory servers are in this category. Only predetermined fields can be used to describe any entity, and they have a centrally agreed upon meaning (within the domain of the original users of the database). This provides semantic homogeneity at the expense of fully flexible expressiveness.

Our research focuses on achieving the middle ground of this continuum: "extensible structured/promoted homogeneity." In particular, our architecture aims to avoid the rigidity of centrally mandated description formats, while minimizing search problems due to "anarchic" descriptions that lack structure or agreed upon vocabularies. Our approach is designed to enable structured descriptions to evolve to meet expanding description requirements for resources, while simultaneously enabling the reuse of metadata.

## 3. COI server architecture requirements

### 3.1. Definitions

This section defines some terms that are used in describing the requirements for the architecture and the remainder of this paper.

*COI Server*: A system that is intended to facilitate the formation and support of Communities of Interest by providing the structured and semantically homogeneous representation of resources

*Object*: The basic unit of information storage within a COI Server. Members of a COI and resources relevant to any COI are all represented as individual objects.

*Objectclass*: The type of an object. It specifies a set of attributes that may or must be present for an object to be of a certain type.

*Attribute-Value Pair*: The atomic element of description of an object. Each object is described by a set of attribute-value pairs. The attribute specifies the meaning of the value that follows.

*User*: Any individual who wishes to discover or provide information using the COI Server

*Member*: An object that is devoted to the description of an individual who may belong to one or more COIs. Members are a subset of users.

*Resource*: Any object contained in the COI Server to help support COIs. This term may include members, but usually refers to information contained in the COI Server other than members (e.g., video camera specifications, FAQs, etc.).

## 3.2. Requirements scenarios

The architecture is designed to achieve the goal of enabling the flexible description of objects that may be of interest to one or more COIs, while promoting semantic homogeneity; these objects include both COI members and resources. To achieve this goal, three broad categories of requirements become evident: it must be possible to create new metadata, it must be possible to describe and discover resources in the COI Server, and, ideally, it should be possible to import existing information from external sources.

Metadata consists of attributes and objectclasses. The architecture must enable a user to define both new attributes and objectclasses to meet the requirement of providing flexible structured descriptions. In addition, since this architecture is aimed at supporting COIs, it is obvious that it must be possible to create new COIs. A less obvious requirement is the ability to promote semantic homogeneity and to reduce semantic heterogeneity within the COI Server after it occurs; since users are free to define new metadata, it is inevitable that synonyms will be introduced. Addressing this requirement allows synonym problems to be rectified.

Resource description and discovery are at the heart of the architecture's goals. To support this, four capabilities are required. Any user must have a mechanism to discover existing description mechanisms (i.e., metadata)

used in the COI Server. The user must also be able to search for resources relevant to a particular information need. In addition, a user must be able to provide new or updated information (i.e., object descriptions) to the COI Server. Finally, a mechanism needs to be provided so users can request to join a COI and be granted official membership.

Table 1. Requirements scenarios

| Scenarios | Name | Purpose |
|---|---|---|
| Metadata Definition | Attribute Definition | Enables a new attribute to be defined |
| | Objectclass Definition | Enables a new objectclass to be defined |
| | COI Definition | Enables a new COI to be defined |
| | Semantic Resolution | Enables the elimination of synonyms in the COI Server by mapping them to a single canonical term |
| Data Access | Metadata Search | Enables existing resource description mechanisms (i.e. metadata) to be discovered |
| | Resource Search | Enables resources of interest to be discovered |
| | Resource Update | Enables the description of a resource to be modified (or the description of a new resource to be created) |
| | Joining a COI | Enables a user to request membership in a COI (or for a user to be signified as a bona fide member of a COI) |
| Translator Access | Replication Definition | Enables a user to specify how to obtain attribute-value pairs for a COI Server object from an external directory |
| | Replication Correction | Enables the elimination of incorrect replication mappings |

Importing information from existing sources is also valuable. For example, a member of a COI may already have information such as his or her email address and telephone number contained within another directory. It is convenient for users to be able to specify the external source of such information since it eliminates redundancy. However, since end users will specify import mappings, a means to correct erroneous mappings is also needed.

These requirements are summarized in Table 1 as a set of specific scenarios that the COI server must support; see Section 1.1 for a motivating real-world example.

Finally, it is also valuable to restrict read and/or update access of particular objects or attributes to certain users. By restricting the read access of certain objects or attribute values to particular users, access to information contained within the COI server can be secured. By restricting the update of certain objects to particular users, the integrity of the description of certain objects is guaranteed by the users who have update access. By restricting the update access of certain attributes to

particular users, the value(s) of those attributes in any object description is guaranteed by the users who have update access to that attribute. However, while these issues are clearly important to a complete implementation of the architecture, the description of security issues is beyond the scope of the current paper.

# 4. An architecture to support COIs

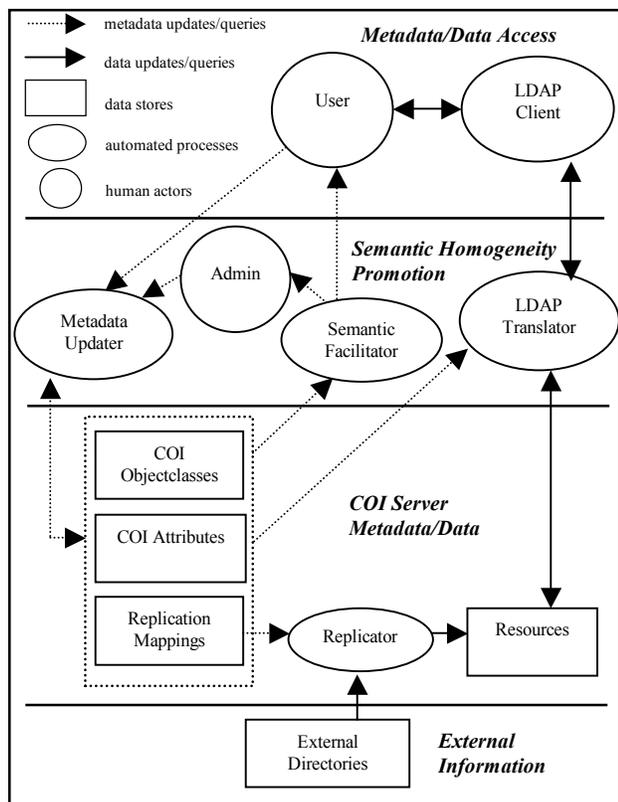## 4.1. Architecture components and sub-components



Figure 2. COI server architecture

The COI Server architecture is shown in Figure 2 and consists of human actors, automated processes and data stores, which are connected to one another through flows of metadata and data. This is consistent with traditional directory services, but goes further by conceptualizing attribute definitions and COI objectclasses as data to be flexibly discovered *and* specified by users, and by providing mechanisms to promote semantic homogeneity.

The architecture has been divided into four major components: External Information, COI Server Metadata/Data, Semantic Homogeneity Promotion, and Metadata/Data Access. The sub-components of each of these components are briefly described below, followed

by a description of how these sub-components operate to satisfy the scenarios described in Table 1.

**External Information**

*External Directories*: Existing sources of attribute-value pairs that describe resources contained within the COI Server.

**COI Server Metadata/Data**

*Replication Mappings*: The specifications of how information contained in an external directory should be imported into the COI Server's description of resources

*Replicator:* The process responsible for using the replication mappings to bring information from external directories into the COI Server.

*Resources:* Any object contained in the COI Server to help support COIs — this includes member descriptions.

*COI Attributes:* The atomic metadata elements used to describe any resource that may be of interest to one or more COIs.

*COI Objectclasses:* A metadata element that specifies the set of attributes that may or must be present in any object describing a resource of a particular type.

**Semantic Homogeneity Promotion**

*Metadata Updater:* The process that enables users to create new COI attributes, COI objectclasses, and replication mappings.

*Semantic Facilitator ™ (SM):* An enhanced interface to enable users or administrators to discover existing metadata relevant to the description of a particular resource. The interface can facilitate presentation of objects to the user with options for clustering objects that are semantically related.

*Administrator:* The individual responsible for the reduction of semantic heterogeneity by determining equivalence among metadata elements, or by correcting replication mappings.

*LDAP Translator:* The process to which any LDAP Client interfaces; this enables every LDAP client to be presented with a semantically homogenous view of the information contained within the COI Server.

**Metadata/Data Access**

*User:* Any individual who wishes to use the system to search for existing resources or to provide fuller description of existing or new resources.

*COI Masters:* Individuals who define new COIs. These individuals are not explicitly represented in the architecture because they are only a special case of users. However, COI Masters have unique privileges to update an attribute signifying that a certain user is a bona fide member of a COI.

*LDAP Client:* A basic LDAP-compatible process that enables users to discover existing data or metadata within the COI Server.

## 4.2. Architecture functionality: scenario walkthroughs

Three groups of scenarios illustrate the operation of the architecture: metadata update, data access, and translator access (see Table 1). These scenarios describe a minimum set of operations required that the COI Server must support to meet its objectives. Without loss of generality, terms from the LDAP directory services standard [1] are used if necessary to illustrate the concepts being described. In addition, for simplicity of exposition, all of the scenarios assume that the hypothetical user has appropriate security clearance to perform the action described. The scenario walkthroughs help in understanding how the architecture (Figure 2) supports the requirements summarized in Table 1.

### 4.2.1. Metadata definition scenarios

Metadata definition is concerned with providing mechanisms to describe resources of a certain type, rather than providing actual data to describe any particular objects. In other words, it is concerned with attribute definitions and objectclass definitions, rather than attribute-value pair specifications or new resource descriptions. Two significant cases of this approach are providing a new attribute definition and providing a new objectclass definition; a special case of this approach is the definition of a COI. For all of these tasks, the user will interface with the metadata updater.

**Attribute Definition**

In this scenario, a user wishes to define a new attribute that can be used to describe resources. This will occur either when a user wishes to provide a more detailed structured description of himself or herself, or wants to provide a mechanism to further describe some resource. The user will first search the existing attribute definitions to determine if an appropriate attribute has already been defined and can be reused; this can be done using either the LDAP client, or preferably, using the Semantic Facilitator™ (SM). If the attribute has already been defined then no further action need be taken; if not, the user will be able to define a new attribute from scratch or by selecting a closely matching attribute definition and modifying it appropriately using the Metadata Updater. In addition, the user can specify an access list of users who have permission to modify values of the attribute. The Metadata Updater is essential, since it enables new metadata definitions to be defined on an open but controlled basis. Once an attribute has been defined, it will be available for use to describe any object in the COI Server.

**Objectclass Definition**

In this scenario, a user wishes to define a COI Objectclass to enable the description of new resources of a particular type. This process is nearly identical to the process of defining a new attribute. First, the user will search the existing objectclass definitions to determine if an appropriate objectclass has already been defined. If so, no further action need be taken; if not, the user will be able to define a new objectclass from scratch or by selecting a closely matching objectclass definition and modifying it appropriately. Once an objectclass has been defined, it will be available for use in creating new objects.

**COI Definition**

A COI is a particularly important entity since it provides the description of members who share a particular interest. However, from the viewpoint of this architecture, the definition of a COI is handled as a special case of objectclass definition and attribute definition. Specifically, for a new COI, a new objectclass is defined that specifies the attribute list appropriate for members of this COI; the user who has created a COI is designated a COI Master.

Two new attributes unique to this COI will be created. The first attribute will be called COI_<COIname>_wantToJoin and will indicate that a user wants to join a particular COI; any user may specify the value of this attribute within his or her own description. The symbol "<COIname>" denotes the name of any particular COI, such as "VideoClub" or "HistoryProfessors". The second attribute will be COI_<COIname>_member, and this will be set to the value of Yes to indicate that a particular individual is a bona fide member of that COI. Updates to the value of this attribute are restricted to those users explicitly named by the creator of a COI.

This approach enables any user to take advantage of the description mechanisms provided by any COI, regardless of whether or not the user is a member of that COI. It also enables individuals who wish to join a COI to indicate this desire using the COI Server's capability to specify an attribute-value pair for an object, i.e. the COI_<COIname>_wantToJoin attribute. Finally, this approach insures that any individual contained in the COI server can be recognized as a bona fide member of a particular COI, since only COI Masters have the ability to set the value of the attribute denoting the COI, i.e. COI_<COIname>_member, to the value of Yes.

**Semantic Resolution**

Ultimately, the resolution of semantic heterogeneity is a human activity. This architecture is intended to promote the semantically homogeneous representation of information. Because the names of all attributes and

objectclasses will always be unique within the COI Server, homonyms cannot occur. However, synonyms can occur. Since synonyms cannot be fully prevented on an automated basis, it is necessary to provide human evaluation and correction of this potential problem.

The recognition of synonyms will derive from the administrator's observation that two or more attribute or objectclass names are being used to describe identical concepts. The administrator will determine a canonical metadata element that "should" be used to describe the particular concept in the COI Server; the "canonical name" for all such metadata elements will be adjusted to name a single metadata element, so that new metadata searches will yield the canonical term. In addition, all existing objects will be adjusted to reflect the new canonical representation.

### 4.2.2. Data access scenarios

Specific data access scenarios are concerned with enabling users to both query and update information contained in the COI Server. In contrast to the metadata update scenarios described above, these scenarios focus on providing the actual values of attributes used to describe particular resources. Four significant cases are contained within this group: (1) a user wants to discover *ways* to search for resources, (2) a user wants to search for particular resources, (3) a user wants to update the description of a resource, (4) a user wants to join a COI.

**Metadata Search**

To perform a structured search, a user must first understand the structure of the information. In this scenario, a user wants to discover the structured description mechanisms, i.e. metadata, existing in the COI Server. The discovery of metadata is supported at two levels through the architecture: (1) through a standard LDAP Client and (2) through the Semantic Facilitator ™ (SM).

All attribute definitions contained within the COI Server are described as particular objects using the predefined COI_attribute or COI_objectclass objectclasses and can thus be discovered using a standard LDAP Client. This lets metadata be discovered simply as data. The definitions of the COI_objectclass and COI_attribute objectclasses are fixed within this architecture. This is analogous to the notion of a SQL database's data dictionary within the ANSI SQL standard [16].

The COI_attribute objectclass provides a data-level description of existing attributes that serve as the building block of COI_objectclass definitions. This information includes the name of the attribute, the syntax of the attribute, a free text description of the attribute's meaning, and a canonical name for the attribute.

The COI_objectclass provides a data-level description of existing objectclasses that are available to users to create new objects. This information includes the name of the objectclass, the schema for the objectclass (i.e., the attributes that must or may be present for a member of the COI), a free text description of the objectclass, and a canonical name of the objectclass.

These objectclasses are well specified because any metadata element will be either a COI_attribute or a COI_objectclass, and no other object within the COI Server will be of those types. In addition, each of the attributes used to describe either of these metadata elements is similarly unique.

Thus, to determine the description mechanisms available within the COI Server, the user will formulate a query for objects that meet a certain condition using the attribute names. Since the attributes describing metadata are all prespecified, this provides a well-defined starting point for a user to discover description mechanisms employed within the COI Server.

The Semantic Facilitator™ (SM) provides an alternative and, ideally, enhanced mechanism for the discovery of metadata and is described in more detail in Section 5. However, the architecture has been designed for improved search capability, without requiring its use.

**Resource Search**

Given the scenario above, a user can find all of the metadata elements that may help to specify relevant objects contained within a COI server. The specification of such relevant objects will be expressed as a query, using a combination of attributes and basic relational operators, combined as necessary using Boolean operators. For example, once a user discovers that the metadata element "published_journals" exists and specifies the journals in which users have published, the user can construct a query such as "published_journals = 'IEEE Computer Software' OR published_journals = 'ACM Computing Surveys'". This will return all members who have indicated publishing in either journal.

**Resource Update**

In this scenario, a user wishes to either update an existing resource description or to create a new resource description. In either case, the operation is essentially the same. Specifically, the user provides a set of attribute-value pairs to uniquely identify the resource, and to then further describe the resource. In the case of an update, it is also possible that the user will specify that certain existing attribute-value pairs be removed.

**Joining a COI**

In this scenario, a user wishes to join a particular COI. To accomplish this, the user will set the value of the "COI_<COIname>_wantToJoin" attribute in his or her member description to Yes.

A COI Master will then be able to find any members who wish to join a particular COI by issuing a query using this attribute. The COI Master will then be able to access any information describing the applicant, and make an adjudication of whether or not the individual should be designated as a bona fide member. If so, the COI Master will add the appropriate "COI_<COIname>_member" attribute-value pair to the member's description, and eliminate the "COI_<COIname>_wantToJoin" attribute-value pair.

### 4.2.3. Translator access scenarios

Translator access scenarios are concerned with two specific cases. The first is enabling users to specify the replication of information contained in external directories into resource descriptions contained within the COI Server. The second deals with an administrator who, using expert knowledge of information contained within the COI Server, works to recognize semantic heterogeneity issues and correct them for the COI Server.

**Replication Definition**

In this scenario, a user wants to make information already stored in an existing directory server available to the COI Server. To accomplish this, the user will define a translation mapping consisting of five elements: external directory, external attribute name, destination attribute, external object key, and COI Server object key.

Given this information, the replicator will be able to access information from the source directory, and replicate the values of attributes in the appropriate COI Server objects. In addition, these translation mappings can be reused, since a user who wishes to map information from an external directory can be advised of any translation mappings from that directory that already exist.

**Replication Correction**

The administrator will be responsible for determining replication mappings that specify attributes from an external directory that do not correspond to the appropriate attribute in the COI Server. At that point, the administrator will simply delete such an erroneous translation mapping, and may introduce a new mapping.

### 4.2.4. Summary

The preceding sections have shown how each of the components and sub-components introduced in the architecture shown in Figure 2 operate in concert to provide the functionality of a COI Server. In addition, given the requirements expressed in Section 3.2, it is believed that these architectural sub-components represent a minimal set of elements to satisfy these requirements.

## 5. COI server prototype

To further evaluate this architecture, we are creating an initial prototype using a single LDAP server. The COI Server thus consists of a set of external processes that provide access to an underlying LDAP Server. Each COI attribute used to describe resources is a first-class attribute (in the LDAP sense), and each COI objectclass corresponds to a first-class objectclass definition within the directory server. This approach has enabled us to provide standard LDAP access to COI information, and to take advantage of the attribute-level and schema-level checking provided by the LDAP server. In addition, all data stores shown in Figure 2 have been implemented using the LDAP Server. The following describes key prototype sub-components: metadata updater, Semantic Facilitator™ (SM), LDAP translator, and replicator.

### 5.1. Metadata updater

Typically, in an LDAP Server, only an administrator has the ability to define new attribute or objectclass definitions. This architecture, however, extends these capabilities to ordinary users. To accomplish this without compromising the integrity of the system, a specialized process, the metadata updater, provides a web-based interface that enables users to request the definition of new metadata elements. A Common Gateway Interface program verifies that the definition is feasible (i.e., correctly specified and not a duplicate), and makes the appropriate LDAP calls to define the attribute or objectclass. In addition, this program creates new objects of the COI_attribute or a COI_objectclass type that correspond to the new attribute or objectclass, so that these metadata definitions can be discovered as standard data elements.

### 5.2. Semantic Facilitator ™ (SM)

The Semantic Facilitator ™ (SM) is probably the most important sub-component in the architecture because it is aimed at promoting semantic homogeneity without requiring human intervention. In this architecture, attributes and objectclasses are seen as data to be discovered by users; given this view, the architecture uses

two of the most successful information retrieval techniques, keyword retrieval and self-organizing maps [17], to enable the discovery of existing metadata. Keyword retrieval is widely used and enables only those attributes or objectclasses that contain certain keywords to be accessed. The results of keyword retrieval are typically presented as a "hit list" showing matching items. However, recent research has indicated that higher-level groups (or clusters) of matching items can improve retrieval effectiveness [18]; this allows the user to see a high-level summary of matching results rather than a long hit list. Self-organizing maps enable metadata elements to be visually presented as groups characterized by a common key phrase; the user can then drill down into any of these groups to view specific metadata elements.

This sub-component is being implemented to support any number of interfaces, such as keyword search only, self-organizing map presentation of results, etc. This will enable users to use a variety of interfaces and select their preferred interface. The prototype allows both searches for metadata elements and the introduction of new metadata elements, and this should enable us empirical study of search or presentation techniques that are successful at promoting semantic homogeneity.

## 5.3. LDAP translator

The LDAP translator is being implemented to more gracefully handle the resolution of semantic heterogeneity by translating the names of attributes and objectclasses to their canonical names.

For example, suppose that a user found an attribute called "frames_per_second" in an earlier interaction with the COI Server and wishes to use that attribute to find the specification of existing video cameras at a later time. Further suppose that since that time, the administrator has determined that "frames_per_second" was actually a synonym for an existing attribute called "fps", and has decided that "fps" should be the canonical term to represent this information. As a result, the administrator will adjust all objects contained within the COI Server to use the "fps" attribute. However, this adjustment would then prevent a query using "frames_per_second" from working properly.

The LDAP Translator automatically handles these issues by acting as an intermediary for all data access requests. The translator receives any requests and replaces any synonyms with their canonical names; it then forwards the request to the underlying LDAP Server. When a response is received, the LDAP translator provides a reverse translation. This enables a user to

transparently use metadata elements discovered in an earlier session.

## 5.4. Replicator

The replicator reduces inter-directory redundancy by enabling the COI Server to retrieve attribute-value pairs describing certain objects from existing external directories and replicate them within the underlying LDAP Server. The current prototype reflects a design decision to use replication of attribute-value pairs (rather than aliases) to provide more consistent performance to COI Server users and greater control of information provided by the COI Server.

The replicator has been implemented using IBM's MetaMerge (http://www.metamerge.com). This tool is middleware designed to enable the replication and synchronization of information among various data repositories including LDAP directories and different RDBMSs. At this point, the replication of attribute-value pairs is being specified as discrete "AssemblyLines" which state the source and destination of information; the updates specified by each "AssemblyLine" are scheduled to occur on a particular interval. While this approach has been able to meet the theoretical goals, we are actively investigating means to optimize updates of information based on external sources to make it more practical.

## 6. Limitations

The most important limitation of this architecture relative to the research objective is the failure to *guarantee* semantic homogeneity for COI and member specifications. The architecture enables full reuse of previously specified attributes and objectclasses, but there is no guarantee that such reuse will occur. This was a necessary tradeoff to maximize expressiveness. However, the architecture's administrator provides a general means by which semantic heterogeneity can be minimized.

No explicit attention has been given to scalability issues. However, the separation of resource descriptions and metadata is our first address to this issue. Specifically, updates to metadata should be relatively infrequent relative to resource updates. This enables metadata to be processed at a central server (and replicated to other servers), and member information to be distributed among multiple servers.

Finally, the architecture described in this paper does not fully address security issues. The architecture and the prototype address key security issues by introducing intermediaries such as the metadata updater that enables users to introduce new metadata or data descriptions,

without the ability to remove existing descriptions. While these measures prevent outright destruction, they also fall short of a comprehensive analysis of needed security measures given the flexibility of the architecture; we are currently working on this issue.

## 7. Conclusions

This paper has proposed a new architecture to support interorganizational collaboration by extending directory services to enable the formation and evolution of groups of individuals who share common interests but work in different organizations. The result has been an approach that retains much of the World Wide Web's flexibility of description, while overcoming its limitations by providing many advantages of centrally mandated and structured descriptions.

We are implementing this prototype and evaluating it against the primary criterion of how well it enables individuals in different organizations to form new COIs and collaborate on research projects. We are empirically evaluating the effectiveness of approaches that increase the reuse of previous specifications of attributes and COIs. While the prototype described is LDAP server based, we are investigating relational databases as a means to analyze resources of existing directories and their semantic heterogeneity issues using data mining techniques.

## Acknowledgments

## References

[1] Tim Howes, Mark Smith and Gordon S. Good, *Understanding and Deploying LDAP Directory Services*, Macmillan Technical Pub., [U.S.A.], 1999.

[2] S. Hayward, J. Graff and N. MacDonald, "Business Strategy Will Drive Directory Services," TG-07-4615, The GartnerGroup, March 11 1999.

[3] Morten T. Hansen, Nitin Nohria and Thomas Tierney, "What's Your Strategy for Managing Knowledge?," *Harvard Business Review*, 77, 2, (1999), pp. 106-116.

[4] Amrit Tiwana and Balasubramaniam Ramesh, "Integrating Knowledge on the Web," *IEEE Internet Computing*, 5, 3, (2001), pp. 32-39.

[5] Abbe Mowshowitz, "Virtual Organization: A Vision of Management in the Information Age," *The Information Society*, 10, 4, (1994), pp. 267-288.

[6] James A. Larson, S. B. Navathe and R. Elmasri, "A Theory of Attribute Equivalence in Databases with Application to Schema Integration," *IEEE Transactions on Software Engineering*, 15, 4, (1989), pp. 449-463.

[7] Sudha Ram, Jinsoo Park, Kangsuck Kim and Hwang Yousub, "A Comprehensive Framework for Classifying Data- and Schema-level Semantic Conflicts in Geographic and Non-Geographic Databases," *Proceedings of the Ninth Annual Workshop on Information Technologies and Systems (WITS '99)*, Charlotte, North Carolina, 1999, pp. 185-190.

[8] J. Damsgaard and D. Truex, "Binary Trading Relations and the Limits of EDI Standards: The Procrustean Bed of Standards," *European Journal of Information Systems*, 9, 3, (2000), pp. 173-188.

[9] Olivier Liechti, "Awareness and the WWW: An Overview," *SIGGroup Bulletin*, 21, 3, (2000), pp. 3-12.

[10] G. Salton, *Automatic Text Processing*, Addison-Wesley, 1989.

[11] Tim Berners-Lee, James Hendler and James Lassila, "The Semantic Web," *Scientific American*, May 2001.

[12] Edward Cannon, *EDI Guide: A Step By Step Approach*, International Thomson Computer Press, London, 1996.

[13] M. R. Gettes and K. Klingenstein, "The Directory of Directories for Higher Education DoD," Internet2 (see http://midleware.internet2.edu/dodhe/), January 2001.

[14] James Tauber, Todd Hay, Tom Beauvais, Mike Burati and Andrew Roberts, "Directory Services Markup Language (DSML)," DSML.ORG, December 2 1999.

[15] Sean McGrath, *XML by Example: Building E-Commerce Applications*, Prentice-Hall, Upper Saddle River, NJ, 1998.

[16] C. J. Date, *An Introduction to Database Systems*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

[17] Teuvo Kohonen, *Self-Organizing Maps*, Springer-Verlag, Berlin, 1995.

[18] Dmitri Roussinov, "Information Foraging through Clustering and Summarization: A Self-Organizing Approach," unpublished Improvement Research, University of Arizona, 1999.