South African Journal
of Information Management
[DoE accredited]

Peer Reviewed Article Vol.2(2/3) September 2000

MySpace: building a dynamic digital persona using directory services

Richard Stubbs
ITD, University of Natal, South Africa
stubbs@nu.ac.za

Contents

1. Introduction

Over the last few years there has been a dramatic increase in the development of Web applications, typically run on a variety of platforms and Web servers. Most systems that provide access to institutional data, or corporate information, maintain a user information database. For example, a Web application that provides access to a card ID system requires login identifications, passwords and user details that are often stored in databases. Likewise a Web application that generates reports for an in-house accounting system contains biographical as well as user authentication information. Medium to large organizations therefore often need to manage a multitude of Web-based information systems and this can result in duplication of user and authentication information. In addition to the databases used in these Web applications, there is also the normal plethora of network access, e-mail and administrative system accounts. The scenario is one of an application-centric approach where users and their details are generated around applications instead of a more user-centric approach where applications are designed around the users and their data. Systems that develop multiple disjointed authentication systems and attempt to manage fragmented information result in user frustration, administrative inefficiency, redundancy and expensive

overheads.

In the next section a number of concepts and technologies are discussed in order to develop a solution to this problem.

top

2. Digital persona

'The digital persona is a model of an individual's public personality based on data and maintained by transactions, and is intended for use as a proxy for the individual' where the digital persona approximates 'personality' (Clarke, 1994). There are many issues surrounding the use of the digital persona, mainly social and socio-legal. Some recent developments have highlighted the use of extra terrestrial programs (ET – adopted from the movie *ET*, where the alien phones home) from the Internet. Such programs accumulate a digital portrait of an individual based on a snapshot of the users personal computer and send it back to a central source. The use of this data is highly circumspect and certainly raises many ethical questions: 'Do I really want any organization to acquire a digital 'picture' of me?' Such ethical dilemmas will not be discussed in this article. Rather, the focus here is on understanding the valid and ethical use of authentic working digital personas managed through well-established organizational network directory services. The networking software company Novell described a persona as 'what makes up your identity is your relationship to things' (Schmitt, 1998), and 'a compilation of the resources and rights relevant to an individual'. For the purposes of this article a digital persona is defined as the identity and public personality of a user, which describes the network and data rights of that user.

top

3. Directory service

Directory services are essentially information sources that contain data describing users, groups, workstations, applications, printers, queues and servers and relationships between the different organizations of data. Examples of directory services are: Novell's Directory Services (NDS) (Novell, 2000), Microsoft's Active Directory Services (ADS) (Microsoft, 1999), Netscape's Lightweight Directory Access Protocol (LDAP) and LDAP itself. Directory services emerged from the initial X.500 project of the late 1980s, an initiative to define and create a global Directory service and namespace.

Directory services provide a replicated and efficient way to store and access different types of data. In addition to being able to replicate the Directory across computers, the directory can be partitioned into manageable chunks. Perhaps one of the most attractive attributes of directory services is the many ways that objects in the directory can be accessed. Access through different protocols and standards (LDAP, Active Directory Services Interface (ADSI) and Extensible Markup Language [XML]) signifies that Directory services can contain almost any data type and in principle can be accessed by diverse clients. An example of this is the inclusion of LDAP search clients in most modern e-mail programs.

The object types in a directory are described by their schema, or metadata. The schema describes the layout of the object type, defining properties or attributes – in essence an object class. The directory is also a hierarchical structure that contains a tree, containers and objects. A container is a special object that contains other objects; it is a container for a group of objects. A tree is a collection of containers and objects in a hierarchical structure. Objects in the Directory are identified by a distinguished name (DN); this name is unique in the tree. A typical name may be defined as:

cn=blogs,ou=Durban,o=nu

where *cn* denotes a common name (e.g. blogs), *ou* an organizational unit (e.g. Durban) and *o* designates an organization (e.g. nu). Nu and Durban are examples of containers and blogs is an example of a user object.

Directory services can also exhibit a feature called inheritance. This is a powerful feature where attributes set at a higher level in the tree can flow down to objects that fall directly under it or into sub-containers. As an example, file access rights could be given to all objects in the container 'Durban' simply by granting those rights to the container Durban. In short a Directory service can:

- Enforce security and access control;
- be adapted to contain new object classes and definitions;
- distribute a directory across computers;
- contain large numbers of objects (due to partitioning and replication); and
- allow easy methods to update and search the directory.

top

4. Lightweight directory access protocol (LDAP)

LDAP was developed as a method to access X.500 directories that were established in the early 90s on typically large Unix computers. The X.500 standard was found to be too overhead intensive to be widely adopted and the LDAP spec emerged to overcome this problem. Originally LDAP was developed by Tim Howes (Howes and Smith, 1995) and a group of colleagues to provide a low cost and easy way to access the University of Michigan's X.500-based directory used to support its e-mail infrastructure (Andreessen 1998). LDAP was released as an RFC in 1993 and the University of Michigan's implementation was provided as freeware. Today LDAP is extensively used as the standard method of accessing and updating directory services. There are stand-alone versions of LDAP that do not provide a way to query a third party directory service but are themselves the directory service. Companies such as Microsoft and Novell heavily push LDAP as a standard way to access and update their respective proprietary directory services. Future directory access mechanisms will leverage on the successful emergence of the XML standard.

Tim Howes (Howes and Smith, 1995) refers to LDAP as 'the universal directory access protocol'. Currently there are task groups and committees developing and reviewing extensions and modifications to the LDAP standard, such as LDAP version 3 (RFC 2251), LDUP (LDAP Duplication and Update Protocol) and LDAPEXT (LDAP Extension). Most LDAP vendors already support LDAP v3.

The LDAP protocol and directory services are therefore universal technological solutions used by network software to access and update network resources. The development of a digital persona therefore necessitates the use of such technologies.

top

5. Environment

The prototype of this project was developed at the University of Natal to enable a Web-based look-up and update facility to the University's directory services. Novell NDS is the directory service that runs at the university. It is used for authentication and user

management, application delivery, e-mail integration, server management, Internet authentication and printing among other things. The directory is spread across three campuses and contains approximately 30 000 objects.

There are a variety of Web servers (Microsoft IIS, Apache and Netscape) running on various platforms (Windows NT, Unix and Netware) at the University. Clients are mostly Microsoft platforms such as Windows 95/98 and NT. The official University Web site (www.nu.ac.za) runs on the Windows 2000 platform using ASP (Active Server Pages) and Microsoft Access databases to provide content. Large portions of this Web site also run on two Sun Solaris Unix boxes located in the Durban and Pietermaritzburg campuses respectively. The University Web content and Web-based applications are spread over three 'virtual services' that seamlessly integrate to provide a single service to users. The University also runs an 'inner Web' server used to supply corporate information to the University community and contains, for example, policy information, guidelines, software and software patches. There are various Web applications that run on different servers and provide information to users based on an identity. Access to such data therefore requires user authorization.

The objective of this project was to develop and evaluate an authentication system for a single Web server (IIS) running on the Windows NT platform. This system necessitated the development of a new object type for the directory services that would control the management and association of Web applications to legal users. The development of such an object required the following:

- A directory tree populated with objects;
- a Web server;
- a Web development environment;
- access methods to the directory, and
- methods to add and update Web application objects.

In the following sections requirements will be discussed.

top

6. Directory tree

The directory tree exists and is used extensively for the registration and use of authorized University users. For the purposes of this project however a test tree was created and populated with 30 000 user objects spread across five containers. Such a tree was created to stress test the system as normal directory design guidelines call for objects to be spread into functional organizational units. Placing 10 000 user objects in one container is not considered best practice, but is done in order to stress test the system. The schema was extended to contain a new object called 'Webbapplication'. The purpose of this object is to define Web applications that a user is able to run. This object is described as follows:

Object	**NU:Webapplication**
Attribute	*Name (string)*
	This is the name of the object and together with the objects context (place in the tree) form the DN (distinguished name) of the object.
Attribute	*URL (string)*
	This attribute contains the URL that the client should be directed to when the link is selected
Attribute	*Image Path (string)*

	This attribute contains the link to an image to display for the application
Attribute	*WebappAssociations (DN List)*
	This attribute contains a list of DN names of objects that can access this application. This object can be users, groups or containers
Attribute	*DescriptionL (string)*
	This attribute contains the description of the application to be displayed on Web Pages; the maximum size of this is 256 characters
Attribute	*DescriptionS (string* – maximum of 45 characters)
	This attribute contains the short description or name that should be displayed in a menu.

The user, group and organizational unit objects in the tree were modified to contain an attribute NU:WebappAssociations which is a DN list of applications that the particular object can run.

top

7. Web server

Owing to current expertise in developing for the Windows environment, Microsoft's IIS running on a NT platform was selected as the Web server. It should be noted that the methods developed here could easily be ported to other platforms and development environments.

top

8. Web development environment

The Web development environment included server-side ASP technology utilizing VBScript and JavaScript. Delphi was used to develop prototype ActiveX controls to wrap the use of common LDAP calls (i.e. read object attribute).

top

9. Access to the directory

Access to the directory is via LDAP ActiveX controls that connect to an LDAP server running on NDS v8.

top

10. Methods to update/add Web objects

The Web application object was developed and prototyped using a Novell tool called SchemaMax. The Web application object administration was therefore accomplished using the Novell Administrator program with a SchemaMax snap-in.

top

11. Process

Firstly a Web-based login page was developed using ASP and client and server side ActiveX

controls that authenticates the user to a LDAP server. Controls used are a shareware control written by 'Polonia Online Information Services' (http://www.polonia-online/ldap/) and ActiveX controls from Novell (http://www.developer.novell.com/). User input is gathered via a Web-based form where username and password are requested. When this page is loaded an ActiveX control is downloaded onto the users PC that returns the NDS username of the user (providing the user has logged onto the directory from his/her PC) to the Web server that then displays it on the login form. If the user is not currently logged into the directory the username field is left blank. The user now types in his/her password (this is the same password as that used to authenticate to the directory) and submits the form by pressing the 'go' button. The Web server has retrieved the username of the user and already knows who the user is; therefore the password is simply used to enforce security and to prevent misuse of the system. To further enforce security, SSL can be used from the client browser right through to the LDAP server.

Once the Web server has received the username and password of the user it logs into the LDAP server via the ActiveX LDAP control as the user (Figure 1) and retrieves common information about the user. This includes first name, surname, e-mail address and an array of information containing appropriate Web applications. These variables are stored in a server session variable and are available for the life of the user's session. Any application running on the Web server will have access to these session variables. This is a single server solution; a multiple server solution is a requisite and is discussed later.

top

Figure 1 LDAP login procedure

```
Username = request.form("Username")
Password = request.form("Password")
Set Ldap = Server.CreateObject("NWIDirLib.NWIDirCtrl.1")
Ldap.FullName = "ldap://196.10.12.1/o=myplace"
LdapUsername = "ldap://196.10.12.1/" + ConverttoLdap(Username)
Ldap.Username = LdapUsername
Ldap.Password = Password
Set Entry = ldap.FindEntry(LdapUsername)
if Err.Number > 0 then
   Session("Loginerror") = "Invalid Username/Password)"
   Set Ldap = Nothing
   Response.Redirect("http://127.0.0.1/intra/getname.asp")
End if
Session("Loginerror") = ""
session("ldapusername") = ldapusername
session("GivenName") = Entry.GetFieldValue("GivenName",False,"")
session("Surname") = Entry.GetFieldValue("Surname",False,"")
session("Username") = Username
 session("Password") = password
Session("E-mail") = Entry.GetFieldValue("E-mail",False,"")
```

Any application that runs on this server can access the users details via the session variables that have been set. Any other user or object attributes that need to be retrieved can be retrieved in an application by accessing a custom developed ActiveX control, namely 'MySpaceX'. This control has a function that takes as an input the DN name of the object to query and the attribute name and will return the value of the attribute. This control logs into the directory as the user and as such will only be able to retrieve attributes that the user has directory service rights to. This means that all access to objects and attributes is controlled by

directory service rights and no special authorization checks need to be done by the Web application. The developed control also has the ability to access the directory as a privileged user and therefore have access to objects and attributes that the user normally would not have.

The login button was placed on the inner Web base page (for this project a copy of the inner Web is used). Once the user has logged on and is authenticated, the Web page adds to its menu system a new option called 'my details'. This option allows the user to query and update his/her personal details as defined by the Web page programmer and NDS permissions. Along with 'my details' the Web server displays menu selection items for every application returned in the Webapplications session object.

top

12. Use of the Web applications object

The integration of various network services, user resources, corporate data and other information into a single functional entity that does not require repeated authentication by user action requires a functional framework that we have named MySpace. Such a system should be intelligent enough to present to a user all relevant and personalized corporate data and information. The use of the Web application object allied with NDS could provide an integrated mechanism for intelligence to be built into distributed systems.

Therefore, the Web application object is an example of how directory services can be used to build a digital identity or 'persona' that provides seamless authentication, identity attribute retrieval and attribute updating capabilities, and thereby provides a means to easily add new applications into MySpace.

The steps involved in the integration of new services into MySpace include:

1. Application development using existing methods to author distributed Web applications and could include ASP, CGI, DLL, Java or Microsoft’s.Net framework;
2. application authorization functions that verify session variables and on failure request the MySpace framework to authenticate the user. Currently this functionality has only been developed for ASP; and
3. create the Web application object in the directory service and populate the attributes with the necessary data, and grant necessary user rights to the application. Here organizational units could be used to grant rights to groups of users.

Therefore applications designed to integrate into the MySpace framework need to contain specific code that supports user authentication that integrates with the Web application methods and protocols. Web-based application developers will need to bind the application into MySpace using supplied functions and methods.

top

13. Multiple server authentication and control

One of the problems with the current implementation of MySpace is the fact that it is restricted to a single Web server. For MySpace to be effective across diverse software and hardware platforms the framework needs to ensure a login-once access-many Web server strategy. At the present time the implementation has not been finalized. However, it is possible to use LDAP to store session particulars in the directory services. It could use the following senario:

1. Log the user on and create a new session object in NDS, this session object will have a unique GUID that is made from the IP Address of the browser a time stamp and the login name of the user.
2. Set a browser cookie (the GUID) that has session life (to only last the length of the current browser session).
3. All retrieval of directory values and authentication checks use the GUID to retrieve the session information from the Directory via LDAP.
4. This GUID will have a time-to-live (ttl) and therefore will expire in the directory after the specified period of time.

top

14. Displaying dynamic information

The digital persona is based on dynamic information that is retrieved from the directory service. MySpace provides a framework for connecting to the directory, reading and updating values. The implementation and use of data are left to the Web site developer. Since the directory service is used as a cornerstone for other services and applications, it provides a centralized up-to-date repository of information. An example of this is a typical e-mail address search page. Since the search page searches the directory, updates to the e-mail systems (such as new users) automatically result in the information being available on the search page. The same holds true for data such as telephone numbers and location details.

An example of an implementation of MySpace is the following:

1. A user logs in; his/her details are noted and written as a session variable.
2. Two applications are displayed in his/her menu, 'Id Card Lookup' and 'Internet billing information'.
3. The user selects 'Id Card Lookup', the server now retrieves the staff number from the directory, connects to a SQL database (where the ID card system resides) and retrieves the user's photo for display.
4. The user now selects 'Internet billing information', the server uses the user's login name (stored in the session variable) to query a database where his/her Internet usage is retrieved and a fee statement is displayed.

top

15. Future

Several areas need to be addressed:

- The entire process needs to be encased in the SSL protocol to ensure security.
- The system login procedure needs to work across multiple servers.
- Common functions need to be developed to check authentication, retrieve and set directory values that will work across different platforms and Web servers (it is suggested that server side java programs be used).
- A more thorough way of checking that a user can run an application needs to be developed than the current method of leaving it up to the application (ISAPI DLLs and integrated authentication to popular Web servers may need to be developed).

Along with the above, technologies such as XML will be investigated and implemented where possible to provide access to objects in the directory.

top

16. Conclusion

MySpace provides a simple way to login, query and set directory service objects. These objects and their attributes can be used to build dynamic Web pages and applications that portray a user's identity. Users can have the ability to update and include their own information in the directory. Such a system provides a unique methodology for creating 'personalized' portals. Information stored in the directory can be used to create default home pages for all users in the directory. Identity management is an important part of a 'Web presence' and most of the tools and data needed to develop these identities already exist. Technologies such as LDAP and XML will allow ubiquitous access to data stores in a directory. Examples of commercial 'identity management software' are Novell's DigitalMe and Ichain (www.novell.com) , Microsoft's Passport (www.microsoft.com) and portal software available from Oblix (www.oblix.com).

top

17. References

Andreessen, M. 1998. Innovators of the Net: Tim Howes and LDAP. http://home.netscape.com/columns/techvision/innovators_th.html.

Clarke, R.A. 1994. The digital persona and its application to data surveillance http://www.anu.edu.au/people/roger.clarke/dv/digpersona.html.

Howes, T. and Smith, M. 1995. A scalable, deployable, directory service framework for the Internet http://www.isoc.org/HMP/PAPER/173/html/paper.html#CONTENTS.

Microsoft. 1999. enterprise identity management within the digital nervous system – strategy white paper http://www.microsoft.com/TechNet/winnt/Winntas/prodfact/eim.asp.

Microsoft. 1999. Planning for a global directory service http://msdn.microsoft.com/library/backgrnd/html/msdn_plannglobalds.htm.

Novell. 2000. Personalizing and customizing Web content utilizing NDS eDirectory at CNN http://www.novell.com/info/collateral/docs/4621088.01/4621088.html.

Schmitt, E. 1998. The new face of networking. A Novell appnote http://developer.novell.com/research/appnotes/1998/january/a1frame.htm.

ISSN 1560-683X
Published by InterWord Communications for the Centre for Research in Web-based Applications, Rand Afrikaans University