This paper was presented at CUMREC 2000, the Higher Education Administrative Technology conference. It is the intellectual property of the author(s). Permission to print out or disseminate all or part of this material is granted provided that the copies are not made or distributed for commercial advantage and that the title and authors of the paper appear. To copy or disseminate otherwise, or to republish in any form, requires written permission from the authors.
A web authorization attribute management system permits delegating web service authorization to persons throughout the university. Some authorized services include, viewing of students grades, accessing commercial software, accessing library external databases from off campus, sending bulk electronic mail to a "dynamic" list of persons, classes, school faculty etc. When the university chooses to use electronic instruction, resources and business then a comprehensive middleware approach is required to manage creation of student/staff resources, and access to those resources. Authorization is evaluated by a rule using attributes that are treated as groups and roles within a group. A comprehensive web system permits delegating authorization decisions to authorities distributed within departments and organizations. Authorization is built on a NetID (Network Identifier) authentication available to every member of the university community. The E-University requires methods of delegating authority electronically.
Middleware for the Electronic-University.
Delegating Authorization on the Web

A more recent version of this report is available at:
http://charlotte.at.nwu.edu/asteiner/educause99.html

Northwestern University started web authorization with student access to their own grades as part of the "Mandarin Consortium". Students used usernames called NetIDs (network identifiers) for getting grades and for accessing their email. Gradually the power of this NetID was expanded to the authorizations agents of a dean to send bulk-email restricted to students and staff of that school. The granting of new passwords for NetIDs was also delegated to certain schools. The SNAP (Simple Network Account Process) also controls creation of accounts on about 20 UNIX and NT hosts. The passwords for all of these hosts are synchronized, (users have the same username and password). Users get accounts on specific machines depending on authorization attributes. A proxy authorization process also allows authorized NetIDs to access library outside vendor databases when using IP addresses outside Northwestern University domain.

The infrastructure to manage these resources requires authentication, authorization and an authorization delegation system. Most of the resources are accessed via the web. The management of these resources is done entirely through the web using delegated authorization. Initially, most account authorization was contained in a flat file that I personally needed to edit for each change. The web management system was created to allow authorization decisions to be delegated to other NetIDs in a way consistent with other authority delegation within at Northwestern University.

The "trust" value of authorization must always be considered. Authentication and authorization at a university represent some form of contract usually between a person and an organization. Student authorization is based on the student enrollment and its relationship. Employee authorization is based on the employment relationship. Note that an authentication/authorization may not be as strong as the relationship. E.G., students can see their grades, but at Northwestern University, they do not take tests based on network authentication, because there are too many ways to deceive the authentication. A student could give his NetID password to another person who would fill in the test answers for him. The system for giving persons a new password must be as strong as the trust required, but they do not need to be stronger. Some universities give persons several passwords for authentication of different strengths.

Browsers automatically save the username and password associated with a specific host/port until the browser is terminated. Thus, it usually appears that a user has "Logged in" to a web service, because web browsers conspire to preserve that fiction.

Once a requestor is authenticated/identified, then the web server asks its authorization system which attributes are required to fulfill the request. The web server formulates a "rule" which it sends to the authorization attribute engine. The authorization engine evaluates the rule and returns and yes or no response. If the authorization check fails, the web server refuses the request with an access forbidden response. The authorization check protocol is the Mandarin CUSSP protocol, which is very simple, yet allows a very complex rule. LDAP (Lightweight Directory Access Protocol) could be used through a SLAPD (Standalone Lightweight Directory Access Protocol) server. SLAPD permits a shell process to access to the SNAP database containing the authorization attributes.
Authorization attributes consist of such things as: student, staff, instructor for class 010102D30061, school WCAS, group "ac" (administer classes). Many of the attributes are derived from Registrar or Human Resources information that is part of the information feeds to SNAP. Some attributes are relatively permanent; others are so transient that their database is rebuilt each night. Some attributes such as "ac" are managed manually through a web interface. Of course, the web interface tests attributes to see whether a requested change is authorized.

A subset of these attributes is communicated to the synchronized NT and UNIX hosts. Synchronized hosts are sent commands to create and delete accounts and to synchronize their password when it is changed. Some of these attributes are mapped to internal groups on the remote hosts, and users on the hosts have group memberships. The account administration software on remote host executes native routines or scripts to create, delete, and alter users. Special exits permit further customization of actions on the remote hosts. This synchronization has been applied to the PeopleSoft SES software, as well as departmental NT and UNIX hosts.

It is important that the interface for management is unobtrusive, and intuitive. Of course, this simplicity is never completely achieved. A relatively simple interface exists to find a group, and then to add members to the group. See: https://www-snap.it-services.nwu.edu/it/snaps/uchgrp.cgi as user mandemo.

Manually managed groups are used to delegate authorization. This delegation may go to several levels.

The "ag" (administer groups) group is the root group, all other groups are delegated authority by that group. The "admina" (administer accounts) group has the authority to delegate account administration to departments. There are some hidden semantics in the account administration server that links departmental accounts with departmental account administrative groups. This is done because members by making members of the Kellogg Graduate School of Management (KGSM) members of the "mngt" group. The adminmngt group can only administer users in group "mngt". In turn the adminmngt group can delegate the "grantmngt" attribute. The "grantmngt" group can reissue passwords for members of the "mngt" group.
The group management interface permits creating new groups, altering group characteristics and adding members to groups. Groups and group memberships all have an expiration date. At the expiration date, the group membership or group with all its memberships expires. Some of the attributes of a group are:

- name -- a unique name of the group
- termination days -- (-9999 means no termination)
- control -- a rule specifying those who can change group membership
- read -- a rule specifying those who can list group members
- description -- a short description of the use of this group
- owner -- a group or NetID who can change group characteristics
- towarn -- an email address to be warned of group/membership termination
- termination action -- notify (member? owner? membership creator? towarn? …)
- creation date
- change date

Group memberships have the following attributes:

- NetID -- of member
- group -- may be a class, automatic group or manual group
- termination days -- cannot exceed life of group
- role -- groups have a few roles which are descriptive (owner, view, control, regular)
- creator NetID -- last NetID altering membership
- creation date
- change date

Rules describe a set of group memberships that must be true:

- ac -- requestor a member of group ac
- ac dc -- requestor a member of group ac or group dc
- ac - dc -- requestor a member of group ac but not group dc
- dc - (ac pulse) -- requestor a member of ac but not a member of either ac or pulse
- ac + dc -- requestor a member of ac and dc
- ac.c -- requestor a member of ac with role c (control)
- ac.v -- requestor a member of ac with role v (view)

A rule is specified by a server to determine authorization of a request. In a sense, the rules are arbitrary and do not need to use the implied meaning of the group. A server could use rule "ac.c" for viewing a page, and use the rule "ac.v" for changing group membership. Although server rules can use arbitrary meanings for groups, the rules should use the published meanings of the group to prevent chaos. In a system with many authorities, keeping communication straight is critical and difficult.

The termination of memberships is the most problematic area of design. I have not yet implemented a termination warning, because I haven't been quite sure what the best policy's are. In almost ever case, I can think of counter policy for notifying someone about termination. However, I am sure that I want groups and memberships to terminate. Groups fall out of usage. There are already hundreds of groups and there need to be termination rules.

The SNAP process has been in development in C++ on HPUX UNIX for about 5 years. The group management functions have been developed in the last 3 years, with the class list...
information being used for about 2 years. Throughout this period there have been two persons working on this and other projects. Each increase in function has broadened the amount of usage on campus.

The ability to delegate authorization is becoming increasingly important at Northwestern University. Web and other client activity is very widespread. The delegation system permits the management of delegation to be distributed to the departments and organizations that can make the authorization. The E-University has a very distributed authorization structure; the electronic systems must support that structure.