General framework for roles and permissions in Dokeos

Written by Frederik Questier
Presented by Frederik Questier @ Dokeos dev meeting 24/08/2005
Input/ideas by Evie Embrechts, Isabel Deprez, Stijn Van Achter, Plone CMS
Creative Commons License by-nc-sa
**Principles / Key concepts**

- General framework, future proof, also for extensions
- Quite optional:
  - Freedom for admins who can delegate their freedom

- Roles
- Permissions
- Localization
- Inheritance
- Work flows
Roles

Roles (and permissions)

- some should be predefined in Dokeos
- can be changed by admin
- can be created by admin
- can be assigned by admin (and teachers if admin wants that)

**Role creation and changing** are just two of the permissions which can be given to certain roles.
Roles

Some roles are global (G), some are local (L):

- G Visitor (anonymous)
- G Authenticated
- L Guest Course Member
- L Official Course Member (enrolled)
- G Student
- L Teaching Assistant
- G Teacher
- L Official Course Teacher (titularis in dutch/latin)
- L Owner (of objects)
- G Admin
Roles

- Each person can have **several roles** (as in real life)

- **Global roles** and their permissions can be changed and assigned by the admins

- **Local roles** and their permissions can be changed and assigned by the teachers or whoever that has the 'change local roles permission'

- Except maybe the (optional) **'Official' roles** which could come from an external administrative database.
Permissions

- View
- Add
- Edit
- Delete
- Sort
- Suggest
- Review/Publish
- Assign Local Roles
- Create (Local) Roles
- Change Permissions of Local Roles
- Change access permissions (visible/unvisible or world/institution/class/owner)

...
All these permissions can be very general, but context sensitive (localized)

- e.g. 'Add' permission can mean
  - Add
    - course
    - groups
    - links
    - documents
    - forum sections
    - forum topics
    - ...

All these permissions could be split up, but localization is the better way to do it.
Inheritance

- Consider Dokeos as an hierarchical (folder) system
- Local permission settings can be inherited from upper folders.
- Inheritance flag on/off

Example

- By default students have only View and Suggest permissions in (root) link folder/tool.
- Teacher can give students the Add/Publish permissions in the folder 'studentlinks' (and therefore in its subfolders).
Assigning permissions

- Permissions are assigned to roles
- Permissions are not assigned to users
  - because one anyway needs roles (Dokeos has since long Student, Teacher and admin roles)
  - because assigning and checking both user and role permissions is difficult

If teachers are assigned (by admin) 'Change permissions' and 'Change roles' permissions:

- Teachers can change the permissions for all students by changing the permissions for their roles.
- Teacher can change the permissions for certain students by assigning them an extra (existing or new) role.

There could be a few predefined 'Powerfull student' roles which could be locally adapted by each teacher.
Conflicting permissions from different roles:
- **positive permissions overrule !**

This works if local roles are used correctly
- examples
  - don't grant Add/Edit/... permissions to global 'Teacher' role (which would be on each course), but to 'Official Course Teacher'
  - don't grant View permission to 'Student' role (which would be on each course) but to 'Official Course Member (enrolled)'
Workflows

Example:

- Student Suggests something (e.g. link)
- The student is Owner of the object and can still Edit it.
- Teacher (or anyone with local review permission) or Deletes or Publishes the link (i.e. gives View permission to other roles)
How to make it easy?

- **Visualisation**
  - On every location (global, course, group?, tool, file, ...)
    - User Permission Matrix or 'ACL' (Access Control List)
    - Roles page, where first a Role is chosen, and then users can be added.
    - Whenever roles and permissions are shown, it should have links to definitions and Permission matrix
    - ...

- **Implementation**
  - Default roles and permissions can be chosen according the current Dokeos settings
  - All tools should try to match the general permissions
  - ...
