

Sistema de permisos de dotLRN

Raúl Morales Hidalgo

Alberto Pesquera Martín

Innova - Sección de Innovación de la UNED

Copyright © 2006 Innova - Licencia GFDL

Informe sobre la estructura y el funcionamiento del sistema de permisos de dotLRN para poder generar nuevos roles, generar permisos, y realizar una nueva interfaz de usuario para nuevos roles y asignación de permisos.

1. Introducción

La administración de permisos y derechos de acceso es uno de los elementos mas complicados en dotLRN ya que se basa en las relaciones de confianza. Hay dos puntos claves:

- **Acceso.** Los usuarios deberían poder acceder a toda la información y documentos necesarios para realizar su trabajo.
- **Restricciones.** Los usuarios no deberían tener acceso a mas información de la necesaria.

Para resolver este problema se introducen tres tipos de permisos de acceso:

- *Perfiles de usuario* (User profiles) que se corresponden con los tipos de miembros de un grupo.
- *Roles* (Roles) que se corresponden con la función que desarrolla un usuario.
- *Jerarquía de usuarios* (User hierarchy) que se corresponde con la estructura de una organización.

Para mejorar este modelo vamos a estudiar que tipos de permisos existen en cada una de las herramientas básicas y que permisos nuevos harían falta para adaptarlo a nuestras necesidades. Observar que tipos de roles hay definidos para cada uno de los tipos de grupo; y la forma en que se pueden asignar los permisos de acceso a cada uno de los tipos de usuario.

2. Tipos de permiso

2.1. Foros

2.1.1. Permisos existentes

admin: Administración de los foros

forum_moderate: Moderar la publicación de mensajes

create: Crear un hilo nuevo

delete: Borrar un mensaje

read: Leer un mensaje

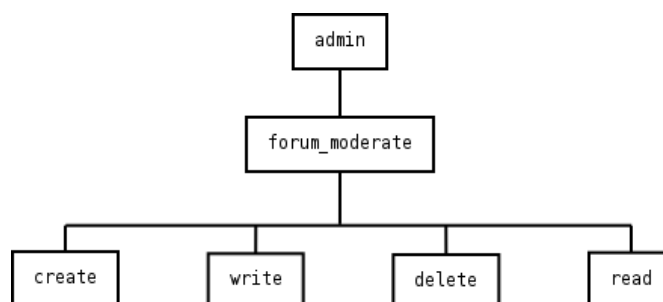
write: Crear un mensaje, publicar una respuesta

En los foros se crea por defecto una estructura jerarquizada de privilegios del siguiente modo:

- Se crea el privilegio forum_moderate como privilegio hijo de admin
- Se crean los privilegios create, delete, read y write como privilegios hijos de forum_moderate

Para crear foros es necesario el privilegio admin en la instancia del paquete dotlrn del tipo de grupo. Create en un forum otorga permiso para crer hilos y write en un mensaje permite publicar respuestas a ese mensaje. Los usos de Read y delete son los obvios.

Figura 1. Jerarquía de permisos



2.1.2. Ampliación

Existen dos ampliaciones posibles, añadir permisos específicos o mejorar la interfaz de adjudicación de permisos.

La primera forma implicaría distinción entre foros e hilos en los permisos, usando los primeros para gestionar los distintos tipos de foros (privados, publicos), y los segundos para gestionar las políticas de publicación de los mensajes dependiendo de los roles.

la segunda implicaría mejorar la gestión de adjudicación de permisos y la creación de roles, permitiendo que cualquier limitación que se quiera hacer se haga creando un grupo específico, asignándoles un rol y aplicando el privilegio que queramos al segmento específico de ese rol.

2.2. Calendario

2.2.1. Permisos existentes

Calendario

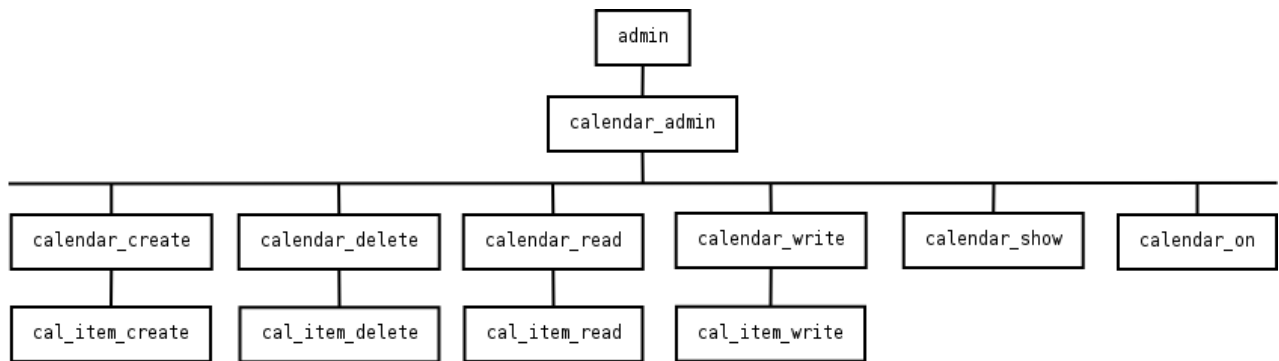
calendar_admin: Administración del calendario
calendar_create: Crear un calendario nuevo
calendar_delete: Borrar un calendario
calendar_read: Leer un calendario
calendar_write: Editar las propiedades de un calendario
calendar_show: Muestra el calendario
calendar_on:

Eventos del calendario

cal_item_invite: Invitar a otras personas al evento
cal_item_create: Crear un nuevo evento
cal_item_delete: Borrar un evento
cal_item_read: Leer un evento
cal_item_write: Editar las propiedades de un evento

No se utilizan los permisos específicos para los elementos del calendario. Sino que por la jerarquía de permisos, se extienden los permisos otorgados al calendario a los propios eventos.

Figura 2. Jerarquía de permisos



2.2.2. Ampliación

Quizas sea una mejora menor, o incluso innecesaria, el definir una interfaz para la gestión de permisos sobre una cita en concreto; ya que gracias la jerarquía de permisos, con los permisos otorgados sobre el calendario podrían bastar.

2.3. Documentos

2.3.1. Permisos existentes

admin: Administrar

create: Crear documentos

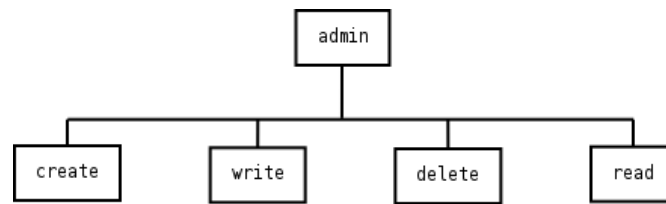
delete: Borrar documentos

write: Modificar documentos

read: Leer

El área de documentos no aplica ningún tipo de privilegio específico, usando los admin, read, write, create de la plataforma.

Figura 3. Jerarquía de permisos



2.3.2. Ampliación

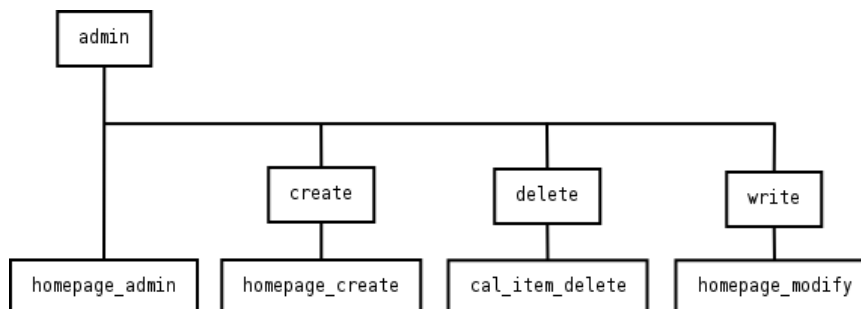
En el caso del área de documentos tampoco es necesario crear mas privilegios, bastaría con una vista de carpeta donde puedas asignar permisos mediante checkboxes, no individualmente como se hace actualmente.

2.4. Homepage

2.4.1. Permisos existentes

- `homepage_admin`: Administrar
- `homepage_create`: Crear elementos
- `homepage_delete`: Borrar elementos
- `homepage_modify`: Modificar elementos
- `homepage_visit`: Acceder al espacio

Figura 4. Jerarquía de permisos



2.4.2. Ampliación

Debería reordenarse la jerarquía de forma que todos los permisos específicos del homepage cuelguen del permiso específico de administración (`homepage_admin`).

3. Roles

Los roles en el sistema de dotLRN son considerados desde un punto de vista lógico. Es perfectamente concebible que un usuario tome uno o mas roles. It is also conceivable that a logical role is represented by multiple physical underlying roles and permissions.

3.1. Cursos

3.1.1. Roles existentes

instructor: Profesor

ta: Profesor asociado

ca: Tutor

cadmin: Administrador del curso

student: Alumno

Un curso ya no es un grupo entre iguales, se pueden distinguir roles con privilegios (profesores, tutores, profesores asociados y administradores del curso) y los alumnos, que es a quienes está enfocado el curso.

3.2. Comunidades

3.2.1. Roles existentes

administrator: Administrador. Usuario registrado encargado de coordinar una comunidad particular.

member: Miembro.

Una comunidad es un grupo de usuarios que trabajan juntos e intercambian varios tipos de información. Puede haber diferentes tipos de comunidades, pero todas tienen unas propiedades básicas. Por lo que las

comunidades deben de ser principalmente, lo mas abiertas posibles; para que luego se vayan definiendo según las necesidades específicas de cada grupo.

3.3. Departamentos

3.3.1. Roles existentes

administrator: Administrador

member: Miembro

Ya que lo que se pretende incentivar es el trabajo colaborativo y cooperativo dentro de la universidad. Estos grupos abogan por una estructura plana y horizontal donde todos son iguales y disponen de los mismos privilegios. Naturalmente alguien se tiene que encargar de algunas labores administrativas, de ahí que exista la figura del administrador.

Y aunque no se han creado los grupos con una idea restrictiva, se pueden otorgar y denegar permisos de acceso y escritura a las distintas herramientas, tal y como disponga el/los administrador/es del grupo.

3.4. Facultad

3.4.1. Roles existentes

administrator: Administrador

member: Miembro

Al igual que los departamentos tienen un fin colaborativo.

4. Segmentos de relación

Para aclarar mucho más las cosas con respecto a los permisos, se usan segmentos de relación y contextos. El objetivo de los *segmentos de relación* es determinar los grupos de usuarios a quién se les conceden los permisos. El objetivo del *contexto* es crear una jerarquía de objetos en la que sus nuevos componentes añadidos, tengan los permisos naturalmente que se extiendan de la manera apropiada.

En OpenACS 4 se ha definido un mecanismo para generalizar los tipos de relaciones. Los tipos de relaciones permiten a los desarrolladores definir mapeos en general para objetos de un tipo T, a otro objeto de un tipo R. Cada tipo de relación es un subtipo de `acs_object`, extendiendo con con atributos extras que guardan las restricciones de la relación. En cambio, cada instancia de un tipo de relación es un objeto que representa un único dato de la forma "el objeto t de tipo T está relacionado con el objeto r de tipo R". Ésto significa que cada instancia de un tipo de relación es esencialmente entre un par de objetos.

Tipos de relaciones

•

Usuarios

- `admin_rel`
- `membership_rel`
- `dotlrn_guest_rel`
- `dotlrn_non_guest_rel`

•

Entre grupos

- `composition_rel`

•

Entre grupos y usuarios

- `dotlrn_admin_rel`
- `dotlrn_external_profile_rel`
- `dotlrn_instructor_rel`
- `dotlrn_professor_profile_rel`
- `dotlrn_student_profile_rel`

5. Gestión de asignaciones

5.1. Configuración por defecto

5.1.1. Foros

Tabla 1. Asignación de permisos por defecto para Foros

	administrar	moderar	crear	borrar	escribir	leer
administrador	Sí	Sí	Sí	Sí	Sí	Sí
miembro	No	Sí	Sí	Sí	Sí	Sí
profesor	No	Sí	Sí	Sí	Sí	Sí
asociado	No	Sí	Sí	Sí	Sí	Sí
tutor	No	Sí	Sí	Sí	Sí	Sí
alumno	No	No	No	No	Sí	Sí

5.1.2. Calendario

Tabla 2. Asignación de permisos por defecto para el calendario

	administrar	crear	borrar	editar	leer	mostrar
administrador	Sí	Sí	Sí	Sí	Sí	Sí
miembro	No	Sí	Sí	Sí	Sí	Sí
profesor	No	Sí	Sí	Sí	Sí	Sí
asociado	No	Sí	Sí	Sí	Sí	Sí
tutor	No	Sí	Sí	Sí	Sí	Sí
alumno	No	No	No	No	Sí	Sí

5.1.3. Documentos

Tabla 3. Asignación de permisos por defecto para el área de documentos

	administrar	crear	borrar	escribir	leer
administrador	Sí	Sí	Sí	Sí	Sí
miembro	No	Sí	Sí	Sí	Sí
profesor	No	Sí	Sí	Sí	Sí
asociado	No	Sí	Sí	Sí	Sí
tutor	No	Sí	Sí	Sí	Sí
alumno	No	No	No	Sí	Sí

5.1.4. Homepage

Tabla 4. Asignación de permisos por defecto para el homepage

	administrar	crear	borrar	modificar	visitar
administrador	Sí	Sí	Sí	Sí	Sí
miembro	No	Sí	Sí	Sí	Sí
profesor	No	Sí	Sí	Sí	Sí
asociado	No	Sí	Sí	Sí	Sí
tutor	No	No	No	Sí	Sí
alumno	No	No	No	No	Sí

5.2. Interfaz de usuario

5.2.1. Introducción

La interfaz de usuario de permisos en dotLRN se divide principalmente en dos páginas, una para gestionar los permisos existentes y otra para crear nuevas asignaciones.

En la primera de las páginas tenemos una matriz de check boxes con los privilegios en las columnas y los segmentos de relación en las horizontales. Esta matriz muestra la asignación de privilegios actual.

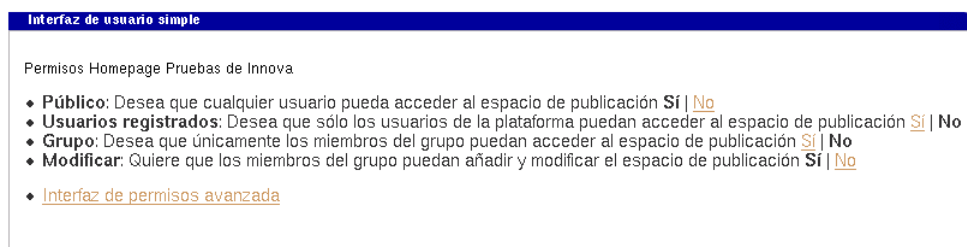
La segunda de las páginas es una página de asignación de nuevos permisos, en la que tenemos dos desplegables, uno para seleccionar privilegios y otro para seleccionar partes o segmentos de relación, siendo el problema aquí que no se limitan los privilegios otorgables a los del paquete sino que salen todos.

La idea es ampliar la usabilidad de estas dos páginas permitiendo una mayor sencillez de interfaz a la vez que se amplía el rango de posibilidades.

5.2.2. Interfaz de usuario simple

Sintetizar los permisos concretos del paquete y mostrar las agrupaciones mas generales, y los ya otorgados. Mostrando una interfaz simple, con una serie de opciones listadas y explicadas brevemente su cometido:

Figura 5. Interfaz de usuario simple



5.2.3. Interfaz de usuario avanzada

Permisos completos mostrados de una forma estructurada.

Figura 6. Interfaz de usuario avanzada

Nombre y apellidos:	Read	Create	Write	Admin	Borrar Todos
pruebasdeinnova	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Administradores de Pruebas de Innova	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Main Site Administrators	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Main Site Members	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Registered .LRN Guests	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Registered .LRN Non-Guests	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
The Public	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Registered Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 7. Interfaz de usuario avanzada 2

