Softwaretechnik-Praktikum SS 2007

Entwurfsbeschreibung –
Rollen- und Rechtemanagement in OLAT

Inhaltsverzeichnis

1. Allgemeines

OLAT (Online Learning And Teaching) ist webbasiertes Learning Management System (LMS). Es nahm seine Anfänge 1999 am Institut für Informatik als PHP Applikation. Mit dem Release 3.0 wurde OLAT komplett neu in Java geschrieben und ist nun ein vielseitiges und, über sogenannte Extensionspoints, gut erweiterbares LMS. Die Zuweisung von Zugriffsrechten innerhalb von OLAT erfolgt durch Zuordnung von Rollen an die einzelnen Benutzer.

2. Produktübersicht

Das Rollen- und Rechtemanagement in OLAT basiert auf den sechs im System vorgesehen Rollen. Jedem OLAT Nutzer wird eine dieser Rollen zugewiesen und damit werden seine Zugriffsrechte festgelegt. Jede Rolle bringt also entsprechende Rechte mit sich. Möglich sind innerhalb von OLAT folgende Rollen:

Gast:
Jeder Nutzer von OLAT, der sich nicht registriert hat, bekommt automatisch diese Rolle zugewiesen. Ein Gast kann nur auf explizit für ihn freigegebene Ressourcen zugreifen. Mit Hilfe der Gastrolle kann ein nicht bei OLAT registrierter Anwender sich also einen Überblick über die angebotenen Kurse und Funktionen verschaffen. Er hat allerdings nicht die Möglichkeit diese Angebote zu nutzen. Die Anmeldung für Gäste kann auch komplett deaktiviert werden.

Benutzer:
Jeder Nutzer der sich OLAT erfolgreich registriert, bekommt automatisch die Rolle "Benutzer" zugewiesen. Deshalb wird im Allgemeinen die Mehrheit der Nutzer von OLAT diese Rolle einnehmen. Als Benutzer steht einem der volle Umfang der im „HOME“ Bereich angebotenen Funktionen zur Verfügung. Ein Benutzer kann also z.B. seinen eigenen Terminkalender bearbeiten, Notizen erstellen, Arbeitsgruppen erstellen (mehr dazu später), Dateien in seinen persönlichen Ordner hochladen, usw.. Außerdem hat er die Möglichkeit sich in die verschiedene angebotene Lernressourcen als Teilnehmer einzuschreiben. Jede weitere Rolle in OLAT baut auf den Rechten eines Nutzers auf und erweitert diese. Deshalb kann die Rolle Benutzer als Basisrolle in OLAT angesehen werden.

Autor:
Zusätzlich zu den Rechten eines normalen Benutzers kann ein Autor auch Lernressourcen erstellen und verwalten. Als Erstellter einer Ressource ist immer auch der Eigentümer dieser und hat volle administrative Rechte innerhalb der Ressource. Er kann neue Kursbausteine einfügen wie z.B. Tests, Fragebögen, Wikis, Glossare, Ressourcenordner, usw.. Der Autor kann Benutzer in die Gruppen einladen bzw. kann es den Benutzer ermöglichen sich selbst einzuschreiben. Der Autor kann auch Benutzer wieder aus den Gruppen entfernen. Er hat auch die Möglichkeit anderen Benutzer zu Eigentümern machen, welche dann für diese Ressource dieselben Rechte hat, wie der Ersteller. Und schließlich kann ein Autor neue Lernressourcen erstellen und

verwalten. Zu Lernressourcen zählen beispielsweise Kurse, Tests, Fragebögen, Wikis, Glossare, Ressourcenordner und SCORM Lerninhalte.

Benutzerverwalter:

Ein registrierter Nutzer mit der Rolle Benutzerverwalter kann nach Benutzern suchen, Einstellungen und Angaben von Benutzern einsehen und ändern, neue Benutzer erstellen und importieren und existierenden Benutzern Rechte und Rollen zuordnen.

Gruppenverwalter:

Ein registrierter Nutzer mit der Rolle Gruppenverwalter kann kursübergreifende Lern- und Rechtegruppen erstellen und verwalten und diese Gruppen mit Kursen verknüpfen. Gruppen, die nicht kursübergreifend sind, können allerdings auch ohne diese Rolle verwaltet werden.

Systemadministrator:

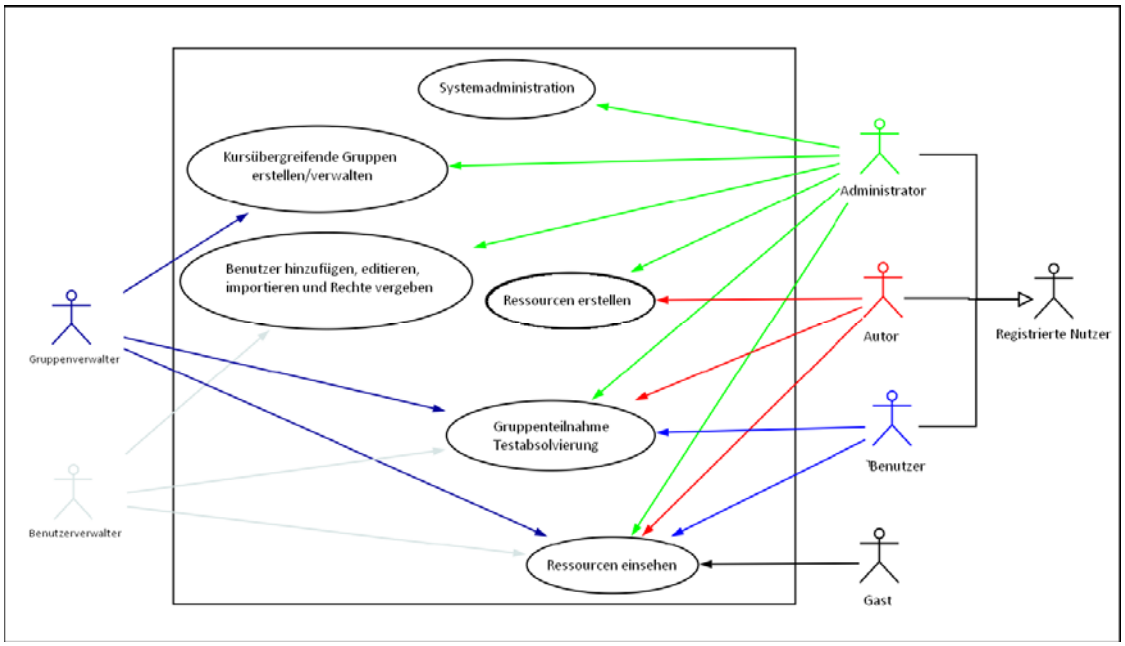
Ein registrierter Nutzer mit der Rolle Systemadministrator hat sämtliche Rechte eines Benutzers, Autors, Benutzerverwalters und Gruppenverwalters. Er verfügt in jedem Kurs bzw. jeder Lernressource über die gleichen Rechte wie die zuständigen Kursadministratoren. Außerdem kann er sämtliche verfügbaren technischen Einstellungen verändern und hat teilweise ein erweitertes Suchformular zur Verfügung.
Hier eine Auswahl der Optionen, die dem Administrator zur Verfügung stehen:

- Auflistung aller momentan eingeloggten Benutzer
- Absenden einer Information für alle Benutzer
- detaillierte Anzeige von Fehlermeldungen (durchsucht das Logfile von OLAT)
- Einstellen der Loglevels der einzelnen Javaklassen

OLAT Systeminformation

- Aktuellen Memoryverbrauch der Java Virtual Machine
- Anzahl concurrent users dispatches zum Zeitpunkt der Anzeige
- die aktuellen Java-Threads
- Java Environment Variablen
- Snoop: Anzeige aller Details einer HTTP-Anfrage
- Usersessions: technisch detaillierte Darstellung der Zustände der einzelnen Tomcat/OLATSessions
- Locks: Hier wird angezeigt welche Benutzer gerade einen Lock besitzen
- Anzeige aller Caches, d.h. aller Objekte, die von Hibernate aus Gründen der Performance im RAM zwischengespeichert wurden
- Quota Verwaltung
- Auflistung der erweiterten Systemproperties eines Benutzers

Zur Übersicht:



Neben den Rollen sind die Gruppen ein weiteres grundlegendes Prinzip in OLAT das die Zugriffsrechte eines Benutzers determiniert. Innerhalb von OLAT gibt es drei Typen von Gruppen (BusinessGroups):

Arbeitsgruppen:

Eine Arbeitsgruppe dient der Gruppenarbeit außerhalb des Kurskontextes. Deshalb ist jeder OLAT Benutzer in der Lage eine Arbeitsgruppe zu erstellen. Er kann seine eigene Gruppe dann mit beliebigen Personen füllen indem er weitere Personen einlädt oder Personen wieder aus seiner Gruppe entfernt. Einzige Bedingung ist, dass die jeweiligen Personen auch OLAT-Benutzer sind. Die Gruppen kann je nach Bedarf mit verschiedenen Werkzeugen, einem Forum und/oder einem Ablageordner (Speicherplatz) ausstattet werden.

Typische Beispiele für Arbeitsgruppen sind:

Studiengruppe:

Studierende können gemeinsam über Lerninhalte diskutieren und Dokumente austauschen.

Autorengruppe:

Autoren bearbeiten gemeinsam ein Dokument und wollen erst die definitive Fassung als Lerninhalt in die Lernressourcen ablegen.

Projektgruppe:

Wissenschaftliche Mitarbeiter arbeiten gemeinsam an einer Publikation.

Lerngruppen:
Als Autor einer Lernressource kann man dort Gruppen erstellen, in die sich die OLAT Nutzer einschreiben müssen oder in die sie eingeladen werden können. Diese mit einem Kurs zusammenhängenden Gruppen werden in OLAT Lerngruppen genannt. Man kann bestimmen, welche und wie viele Mitglieder die Gruppe hat und den Gruppen Betreuer zuweisen. Betreuer können die Lerngruppe administrieren d.h. Mitglieder in die Gruppe aufnehmen und weitere Betreuer ernennen. Mehrere Lerngruppen können zu einem Lernbereich zusammengefasst werden. Kursübergreifende Lerngruppen können nur von Nutzern mit der Rolle Gruppenverwalter" erstellt und verwaltet werden.

Rechtegruppen:
Als Autor einer Lernressource kann man anderen OLAT-Benutzern bestimmte Rechte an dieser erteilen indem man diese Benutzer in eine Rechtegruppe einlädt. Man kann alle OLAT-Benutzer in eine Rechtegruppe aufnehmen unabhängig von deren bisherigen Rollen in OLAT. Die Rechte, die man einer Rechtegruppe zuweisen kann, sind:

- das Recht Lerngruppen zu managen
- das Recht einen Kurs zu editieren
- das Recht Kursdaten zu archivieren
- das Recht andere Kursteilnehmer zu bewerten

Zusammenfassend kann man die Funktionalität der Gruppen in OLAT so beschreiben:

Lerngruppen werden in Lernressourcen verwendet um Personen aus administrativen oder pädagogischen Gründen zu gruppieren, z.B. um Ihnen Lernmaterialien in einem geschützten Bereich zugänglich zu machen.
Rechtegruppen werden in Lernressourcen verwendet um Personen gezielt spezielle Rechte innerhalb eines Kurses zuzuteilen, z.B. um die Arbeit aufzuteilen.
Arbeitsgruppen können von allen Benutzern selbst erstellt werden um z.B. gemeinsam an einem Projekt zu arbeiten oder Dokumente auszutauschen. Sie stehen in keinem direkten Zusammenhang mit einem Kurs.

3. Grundsätzliche Struktur und Entwurfsprinzipien für das Gesamtsystem

Das Rollensystem von OLAT besteht aus den sechs Rollen: Gast, Benutzer, Autor, Benutzerverwalter, Gruppenverwalter und Systemadministrator. Dabei sind Benutzer, Autor, Benutzerverwalter, Gruppenverwalter und Systemadministrator die angemeldeten Systembenutzern. Die Rollen der Systembenutzer bauen teilweise aufeinander auf so ist zum Beispiel jeder Autor, Benutzerverwalter, Gruppenverwalter und Systemadministrator auch ein Benutzer. Der Systemadministrator erhält die Rechte aller anderen Systembenutzerrollen, ist jedoch nur zur technischen Wartung von OLAT vorgesehen. Jeder angemeldete Akteur besitzt automatisch die Rolle Benutzer, kann aber durch den Administrator (oder Benutzerverwalter) von den Systembenutzerrollen befreit werden und als anonymer Nutzer (der dann die gleichen Rechte wie ein Gast hat und dieser Rolle entspricht) eingestuft werden. Anhand der Rollen wird eine grobe Unterteilung durchgeführt, welche Seiten dem Benutzer zur Verfügung stehen. Diese sechs Rollen sind ausreichend um die Funktionalität aller OLAT spezifischen Aktionen zu kontrollieren.

Es besteht jedoch die Möglichkeit zusätzliche Rollen in den entsprechenden Tabellen zu erzeugen und in einer Extension separat abzurufen und zu nutzen, ohne das dabei die Grundfunktionen beeinflusst werden. Dem authentifizierten Akteur werden dabei die zusätzlichen Mitgliedschaften zugeordnet (zum Beispiel in einem zusätzlichen Administratrationswerkzeug).

Zusätzlich zum Rollenmanagement gibt es in OLAT das Rechtemanagement. Dieses ermöglicht, dass einzelne Seiten oder Funktionen für bestimmte Benutzer gesperrt beziehungsweise freigegeben werden. Dafür wird das entsprechende Zugangsrecht mit dem Behälterobjekt verknüpft. Um gesonderte Rechte auf einen Behälter zu erhalten, muss dieses explizit erteilt werden.

Das Rechte-Konzept von OLAT orientiert sich stark an dem Policy-Konzept von Java. Es gibt:

- Identities (User)
- Gruppen mit Identities darin sogenannte Securitygroups (nicht Businessgroups)
- Rechte
- Resourcen / Objekte
- Policies

User sind in einer oder mehreren Gruppen. Eine Policy ist die Kombination von einer Gruppe, einem Recht, und einer Resource. Ein Benutzer hat dann ein gewisses Recht (z.B. lesen) auf einen gewisse Resource (z.B. RepositoryEntry), wenn er in mindestens einer Gruppe ist, die von einer Policy referenziert wird, welche das Recht „lesen“ auf die Resource hat, d.h. die Policy hat einen Fremdschlüssel auf die Gruppe, einen Fremdschlüssel auf die OLAT-Resource, und einen Text für das Recht. Die Rechte sind additiv und positiv:

additiv: Alle Rechte von allen möglichen Gruppen, in denen ein User ist, werden zusammengezählt, und ergeben so als Summe die Rechte dieses Benutzers

positiv: Alle Rechte beschreiben die Erlaubnis, etwas zu tun (Im Unterschied zu negativen Rechten, die explizite Verbote beschreiben)

Die Systemrollen sind demzufolge nichts anderes als Gruppen mit bestimmten Rechten.

Da OLAT ein System mit vielen Anwendern ist, sind sowohl Rollen als auch Rechtemanagement nötig um jedem Nutzer die von ihm benötigten Funktionen zur Verfügung zu stellen. Durch die Zweiteilung wird

erreicht, dass der Datenbestand, der die Benutzerrechte repräsentiert, so gering wie möglich gehalten wird.

4. Grundsätzliche Struktur und Entwurfsprinzipien der einzelnen Pakete

Org.Olat.Basesecurity:

Das Rollen- und Rechtemanagement von OLAT ist vor allem aus den Objekten des Paketes *org.olat.basesecurity* aufgebaut. Die Klasse *Policy* enthält dabei das eigentlich Recht, was mittels der Methode *getPermission()* abgefragt wird. Die Klasse *PersistingManager* übernimmt beim Rechtemanagement die Rolle des Controllers. Zum Beispiel die Rechte der Rolle Administrator:

```
private void initSysGroupAdmin() {
    SecurityGroup adminGroup = findSecurityGroupByName(Constants.GROUP_ADMIN);
    if (adminGroup == null)
        adminGroup = createAndPersistNamedSecurityGroup(Constants.GROUP_ADMIN);

    // we check everthing by policies, so we must give admins the hasRole
    // permission on the type resource "Admin"
    createAndPersistPolicyIfNotExists(adminGroup, Constants.PERMISSION_HASROLE,Constants.ORESOURCE_ADMIN);

    //admins have role "authors" by default
    createAndPersistPolicyIfNotExists(adminGroup, Constants.PERMISSION_HASROLE, Constants.ORESOURCE_AUTHOR);

    //admins have a groupmanager policy and access permissions to groupmanaging tools
    createAndPersistPolicyIfNotExists(adminGroup, Constants.PERMISSION_HASROLE, constants.ORESOURCE_GROUPMANAGER);

    //admins have a usemanager policy and access permissions to usermanagement tools
    createAndPersistPolicyIfNotExists(adminGroup, Constants.PERMISSION_HASROLE, Constants.ORESOURCE_USERMANAGER);

    //admins are also regular users
    createAndPersistPolicyIfNotExists(adminGroup, Constants.PERMISSION_HASROLE, Constants.ORESOURCE_USERS);

    //olat admins have access to all security groups
    createAndPersistPolicyIfNotExists(adminGroup, Constants.PERMISSION_ACCESS, Constants.ORESOURCE_SECURITYGROUPS);

    // and to all courses
    createAndPersistPolicyIfNotExists(adminGroup, Constants.PERMISSION_ADMIN, Constants.ORESOURCE_COURSES);

    createAndPersistPolicyIfNotExists(adminGroup, Constants.PERMISSION_ACCESS, OresHelper.lookupType(SysinfoController.class));
    createAndPersistPolicyIfNotExists(adminGroup, Constants.PERMISSION_ACCESS, OresHelper.lookupType(UserAdminController.class));
    createAndPersistPolicyIfNotExists(adminGroup, Constants.PERMISSION_ACCESS, OresHelper.lookupType(UserChangePasswordController));
    createAndPersistPolicyIfNotExists(adminGroup, Constants.PERMISSION_ACCESS, OresHelper.lookupType(UserCreateController.class));
    createAndPersistPolicyIfNotExists(adminGroup, Constants.PERMISSION_ACCESS, OresHelper.lookupType(QuotaController.class));
    createAndPersistPolicyIfNotExists(adminGroup, Constants.PERMISSION_ACCESS, OresHelper.lookupType(GenericQuotaEditController));
}
```

Also gibt es eine 1:1 Verknüpfung zwischen einem Recht und einer OLAT-Ressource, welche mittels der Klasse *PermissionOnResourcable* realisiert wird. Dadurch können einzelnen OLAT-Ressourcen unter Benutzern Berechtigungen zugewiesen werden.

So zum Beispiel im *UserAdminMainController*:

```
if (uobject.equals("coauthors"))
{
	activatePaneInDetailView = "edit.uroles";
	// special case: use user search controller and search for all users that have author
	// rights
	PermissionOnResourceable[] permissions =
	{
		new PermissionOnResourceable(Constants.PERMISSION_HASROLE,Constants.ORESOURCE_AUTHOR)
	};
	UsermanagerUserSearchController myCtr = new UsermanagerUserSearchController(ureq, getWindowControl(),null,
	permissions, null, null, null);
	// now subtract users that are in the author group to get the co-authors
	Manager secMgr = ManagerFactory.getManager();
	SecurityGroup[] secGroup =
	{
		ManagerFactory.getManager().findSecurityGroupByName(Constants.GROUP_AUTHORS)
	};
	List identitiesFromAuthorGroup = secMgr.getIdentitiesByPowerSearch(null, null, null,null, null, null, secGroup , null,
	null, null, null);
	myCtr.removeIdentitiesFromSearchResult(ureq, identitiesFromAuthorGroup);
	contentCtr = myCtr;
	contentCtr.addControllerListener(this);
	return contentCtr.getInitialComponent();
}
```

Verwendung findet das Rechtemanagement unter anderem an den folgenden Stellen bei OLAT:

Org.Olat.ControllerFactory:

Die initiale Rollenunterscheidung findet bei OLAT in der *ControllerFactory* statt. Dort wird je nach U*serRequest* die darzustellende Seite bestimmt. Hierbei werden die Rollen zunächst mittels des Befehls *ureq.getUserSession().getRoles()* abgefragt. Je nach der zurückgegebenen Rolle und entsprechend der Benutzeranfrage, gibt dann die Methode *createLaunchController(...)* folgende *Controller*-Objekte zurück:

- *HomeMainController*
- *GuestHomeMainController*
- *SystemAdminMainController*
- *UserAdminMainController*
- *RepositoryMainController*
- *BusinessGroupMainRunController*
- *BGMainController*
- *BGContextManagementController*
- *LaunchController*

So wird zum Beispiel der Zugriff auf den *UserAdminMainController* nur gewährt, falls *roles.isUserManager()* bzw. *roles.isAdmin()* wahr ist. Die anderen Controller werden analog überprüft.

Wird im UserRequest ein Controller angefordert, auf den der Benutzer keinen Zugriff hat, so wird eine OLATSecurityException geworfen.

Beim erstellen eines weiteren Controllers, wie zum Beispiel des *UserCreateController*'s, wird bei der Erzeugung eines Objektes durch folgenden Code abgefragt ob die Voraussetzungen erfüllt sind, um einen Controller zu erstellen:

```
Manager mgr = ManagerFactory.getManager();
if (!mgr.isIdentityPermittedOnResourceable(ureq.getIdentity(),Constants.PERMISSION_ACCESS,OresHelper.lookupType(this.getClass())))
{
throw new OLATSecurityException("Insufficient permissions to access UserCreateController");
}
```

Analog wird der Zugriff auf zugriffsbeschränkte Methoden überprüft. Dies funktioniert auf allen Zugangsbeschränkten Controllern gleich.