TRAUNER VERLAG
UNIVERSITÄT

Schriftenreihe E-Learning

Herausgegeben von
Jörg R. Mühlbacher
Günter Pilz
Bernad Batinic

Jörg R. Mühlbacher ■
Andreas Putzinger (Hg.)

Die WeLearn Plattform

Impressum

Schriftenreihe E-Learning

Jörg R. Mühlbacher ■ Andreas Putzinger (Hg.)
Die WeLearn Plattform

Herstellung:
Kern: Johannes-Kepler-Universität Linz, A 4045 Linz-Auhof
Umschlag: TRAUNER Druck GmbH & Co KG, A 4020 Linz, Köglstraße 14

ISBN 3-85499-061-8
ISBN 978-3-85499-061-1
www.trauner.at

E RECHTESYSTEM IN WELEARN

Ein Ziel beim Design von WeLearn war es, die Lernumgebung möglichst flexibel zu gestalten. Dies betrifft einerseits die möglichen Rollen von Benutzern des Systems im Lern- und Lehrprozess, als auch die Strukturierung und Organisation der Kursinhalte andererseits.

Um diesen Anforderungen in der Praxis gerecht zu werden, ist es notwendig, eine umfassende Rechteverwaltung zu unterstützen. Damit kann sehr feingranular festgelegt werden, welcher Benutzer welche Aktionen auf die einzelnen Objekte im System ausführen darf. Somit ist es beispielsweise möglich, dass ein und derselbe Benutzer für einen Kurs A die Rolle eines Lernenden inne hat, in Kurs B aber Lehrender ist.

Da WeLearn plattformunabhängig ist, kann nicht auf die Rechteverwaltung des darunter liegenden Betriebssystems zurückgegriffen werden. Diese ist beispielsweise unter Windows völlig unterschiedlich konzipiert wie unter UNIX-Derivaten. Daher wurde ein eigener plattformunabhängiger *„Rechtemanager“* in WeLearn implementiert. Dieser ist konzeptionell an Paradigmen angelehnt, welche auch innerhalb des Windows Betriebssystems, wie z. B. bei „NTFS“, verwendet werden. Dies hat unter anderem den Vorteil, dass die Rechte-Administration von WeLearn meist geringen Einarbeitungsaufwand bedeutet, obwohl gleichzeitig hohe Flexibilität und zahlreiche Möglichkeiten zusätzlich geboten werden.

Zum Einsatz kommen so genannte Zugriffslisten (*Access Control Lists,* kurz *„ACLs“*), mit welchen für jedes einzelne Objekt in einem Verzeichnisbaum festgelegt wird, welcher Benutzer welche Rechte auf das besagte Objekt besitzt, d. h., welche Aktionen er darauf durchführen darf. Zusätzlich ist ein Konzept der Rechtevererbung implementiert, um die praktische Nutzung zu vereinfachen. Darauf wird später in diesem Kapitel noch im Detail eingegangen.

E.1 Benutzerkonzept

Für jeden „physischen“ Benutzer eines WeLearn-Systems sollte idealerweise ein eigener entsprechender WeLearn-Account eingerichtet werden. Die genauen Schritte hierfür sind im WeLearn-Administrationshandbuch [#Wel/2] zu finden.

E.1.1 Spezielle Benutzer in WeLearn

Eine Neuinstallation von WeLearn stellt eine vorkonfigurierte Systemumgebung zur Verfügung, in welcher bereits wichtige Verzeichnisse, Benutzer, Gruppen und andere Objekte vorhanden sind. Von den vier voreingerichteten Benutzern sind drei speziell erwähnenswert, da diese Sonderrollen einnehmen und daher auch nicht gelöscht werden können. Dies sind *„everybody“*, *„system“* sowie *„owner“*. Der vierte Benutzer, *„admin“*, wird für die Erstkonfiguration und auch Administration des Systems verwendet. Das Standardpasswort „admin“ sollte aus Sicherheitsgründen unbedingt und unmittelbar nach dem ersten Login geändert werden!

E.1.1.1 Benutzer „everybody“

Um Teile des Systems ohne vorhergehenden Login zugänglich machen zu können, agiert jeder Benutzer, der noch nicht in WeLearn eingeloggt ist, im System als *„everybody“*. Die öffentlichen Teile, die ohne vorherigen Login erreicht werden können, sind direkt im Top-Menü verfügbar. Weiters müssen entsprechende Lese- und Ausführungsrechte für den Benutzer *„everybody“* auf diese Objekte definiert werden.

Mit diesem Konzept können bei Bedarf unkompliziert Dateien, Foren und andere WeLearn-Objekte für alle Internet-Benutzer direkt und ohne Login zugänglich gemacht werden.

E.1.1.2 Benutzer „system“

Ein weiterer spezieller Benutzeraccount ist *„system“*. Dieser wird von der WeLearn-Applikation selbst benötigt, um Funktionen und Anpassungen während der Einrichtung und der Updates von WeLearn durchführen zu können.

E.1.1.3 Benutzer „owner“

Jedes WeLearn Objekt hat einen Besitzer, den so genannten *„owner“*. Hierbei handelt es sich um denjenigen Systembenutzer, der dieses Objekt erstellt hat. Der Besitzer von voreingerichteten Objekten, wie beispielsweise den Ordnern „`/groups`“ oder „`/system`“ ist der Benutzer „system“. Vergibt man nun auf ein Objekt für den Benutzer *„owner“* Rechte, erhält die Rechte effektiv derjenige Systembenutzer, der das Objekt erstellt hat. Somit ist der scheinbar physisch angelegte Benutzer *„owner“* nur ein Konzept, um den jeweiligen tatsächlichen Besitzer eines Objektes indirekt ansprechen zu können, ohne aber im Konkreten für jedes Objekt wissen zu müssen, welcher Systembenutzer dies ist.

E.1.2 Hinweise zu den voreingerichteten Accounts

Obwohl *„everybody“*, *„system“* und *„owner“* als eigene Accounts und somit quasi als Benutzer im System angelegt sind, ist ein Einloggen mit diesen Identitäten unmöglich, da der Login für diese Benutzer standardmäßig deaktiviert ist. Diese Einstellung darf auf keinen Fall geändert werden.

Das Konzept, einzelnen Systembenutzern Rechte auf Objekte zu geben, ist sehr feingranular. Da dies bei einer großen Anzahl an Benutzern schnell unüberschaubar und aufwändig werden könnte, wurde das Konzept von „Gruppen“ implementiert, welches im folgenden Kapitel vorgestellt wird.

E.2 Gruppenkonzept

Um nicht für jeden Benutzer individuell Rechte auf Objekte definieren zu müssen, hat es sich als praktikabel erwiesen, Benutzer in Gruppen zusammenzufassen. Jeder Benutzer, der in einer bestimmten Gruppe Mitglied ist, besitzt logische Gemeinsamkeiten mit den anderen Gruppenmitgliedern. Einzelne Benutzer können in WeLearn in beliebig vielen Gruppen Mitglied sein, wodurch deren Rechte im System in den meisten Fällen kumuliert werden (*Ausnahme:* Deny, siehe Abschnitt E.3.2). Anstatt für individuelle Benutzer Rechte auf ein Objekt zu vergeben, ist es komfortabler, gleich für eine ganze Gruppe, und damit für alle deren Mitglieder, Rechte zu vergeben.

E.2.1 Spezielle Gruppen in WeLearn

In WeLearn existieren zwei spezielle Gruppen, die beim ersten Systemstart automatisch eingerichtet werden. Diese sind *„Users“* einerseits, und *„Administrators“* andererseits. Alle WeLearn Benutzer müssen in genau einer der beiden Gruppen Mitglied sein (vgl. Abschnitt E.3.2), was sie entweder als Systemadministrator oder als Standardbenutzer qualifiziert. Zusätzlich können sie in beliebig vielen weiteren Gruppen Mitglied sein, wodurch beispielsweise Zugriff auf einzelne Kurse gewährt wird.

E.3 Access Control Listen – ACLs

E.3.1 Aufbau

Jedem WeLearn-Objekt ist direkt oder indirekt eine ACL zugewiesen. Diese definiert exakt, welche Benutzer und welche Gruppen welche Rechte auf dieses Objekt haben. In Abb. A-1 ist der Aufbau einer solchen ACL dargestellt. Rechte können also sowohl direkt für einzelne Benutzer vergeben werden, als auch für Gruppen. Diese Rechte wirken dann indirekt wieder auf alle Benutzer, welche in diesen Gruppen Mitglied sind. Dadurch können Widersprüche entstehen, deren Auflösung in Kapitel E.3.2 behandelt wird.

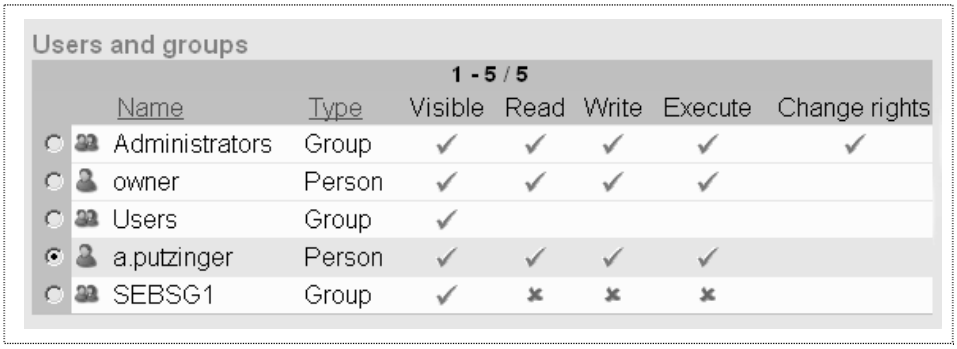

Abb. E-1 *Rechtetabelle*

Für die meisten WeLearn Objekte sind fünf Rechte verfügbar, welche in der ACL auf drei verschiedene Zustände gesetzt werden können. In Messageboards und Kalendern sind sechs Rechte verfügbar, in Foren sieben.

In WeLearn gibt es zwei Möglichkeiten, wie die zu einem Objekt gehörende ACL ermittelt werden kann. Entweder ist dem konkreten Objekt eine eigene ACL zugeordnet, die dann für das jeweilige Objekt gültig ist, oder aber die ACL wird von einem übergeordneten Objekt (meist ein Verzeichnis) übernommen, wobei man hierbei allgemein vom Vorgang des „Vererbens“ spricht.

E.3.2 „Sie haben Recht(e)!“ – Auswertung von Rechten in WeLearn

Bis jetzt wurde davon gesprochen, dass ein Benutzer bestimmte Rechte auf Objekte besitzen kann. In WeLearn gibt es drei Zustände, auf die ein solches „Recht“ gesetzt sein kann. Dies sind *„Grant“*, *„Deny“* und *„Not Set“*. Als erste Faustregel vorweg gilt, dass nach Möglichkeit nur die Zustände „Grant“ und „Not Set“ verwendet werden sollten. Der Grund hierfür liegt in der Strategie, wie und mit welcher Priorität WeLearn die gültigen Rechte auswertet.

Jedes Mal, wenn eine Aktion auf ein Objekt ausgeführt werden soll, wird vom WeLearn-System geprüft, ob diese auch tatsächlich vom aktuellen Benutzer ausgeführt werden darf. Dazu wird zuerst diejenige ACL bestimmt, welche für das besagte Objekt gültig ist. Hat das Objekt eine eigene Zugriffskontrollliste, kann diese direkt verwendet werden. Ist für das Objekt keine eigene gesetzt, so wird die Vererbung aufgelöst. Dazu wird in der Hierarchie nach oben gegangen und geprüft, ob für das übergeordnete Verzeichnis eine eigene ACL definiert ist. Ist dies der Fall, wird diese verwendet, ansonsten wird wiederum eine Ebene nach oben gegangen, solange, bis eine gesetzte ACL gefunden wird. Da auf das so genannte Root-Verzeichnis „/“ implizit eine vom System vorgegebene und nicht veränderbare ACL definiert ist, hat die Suche spätestens dort ein Ende.

Bei der Bestimmung, ob ein Benutzer ein bestimmtes Recht auf ein Objekt hat, werden folgende Punkte miteinbezogen:

- Das *Objekt*, auf welches der Zugriff erfolgen soll.
- Das *Recht*, welches geprüft werden soll. Welche Rechte es in WeLearn gibt, wird in E.4 erläutert.
- Diejenige *ACL*, welche für das Objekt gilt.
- Der *Benutzer*, für welchen die Rechteprüfung erfolgen soll. Ist der Benutzer momentan nicht eingeloggt, wird vom System für die Prüfung „everybody“ verwendet.
- Die Gruppen, in welchen der Benutzer Mitglied ist.

Ist in einem Eintrag in der ACL weder explizit ein „Grant“ gesetzt, noch explizit ein „Deny“, so ist der Zustand „Not Set“. Die eigentliche Überprüfung der Rechte funktioniert wie folgt:

1. *Auswertung nach Besitzer*

   Zuerst wird überprüft, ob der aktuelle Benutzer der Besitzer, also der *„owner“*, des Objektes ist. Ist dies der Fall, wird als nächstes überprüft, ob ein explizites „Deny“ oder „Grant“ gesetzt ist. Ist dies der Fall, steht das Ergebnis der Rechteprüfung bereits in diesem Schritt fest und die Auswertung kann beendet werden. Die Rechte des Besitzers sind somit am stärksten.

   Im Fall, dass kein expliziter Eintrag in der ACL für den Owner existiert oder dass im Eintrag für das zu prüfende Recht weder ein „Grant“ noch ein „Deny“ gesetzt ist, also „Not Set“ gültig ist, wird mit Schritt zwei fortgesetzt.

   In der praktischen Anwendung machen „Grants“ für den Owner vor allem bei Abgabeordnern Sinn, wohingegen es für „Deny“-Einträge für den Owner wohl kaum Anwendungsfälle gibt.

2. *Auswertung nach Benutzer*

   Ist der aktuelle Benutzer nicht der Besitzer des Objektes bzw. ist kein expliziter Eintrag für den Besitzer gesetzt, dann wird als nächstes überprüft, ob für den aktuellen Benutzer ein Eintrag in der ACL vorhanden ist. Ist das der Fall, wird erneut geprüft, ob es eine explizite „Deny“ oder „Grant“ Einstellung gibt. Wenn ja, dann stellt diese das Ergebnis der Rechteüberprüfung dar, und die Auswertung wird nach diesem Schritt beendet.

   Ist für den Benutzer zwar ein Eintrag in der ACL vorhanden, aber für das zu prüfende Recht „Not Set“ konfiguriert, wird die Auswertung bei Schritt drei fortgesetzt.

3. *Auswertung nach Gruppen*

   Wenn bis zum dritten Schritt kein Ergebnis vorliegt, werden noch alle Gruppen überprüft, in denen der Benutzer Mitglied ist. Hierbei gilt allgemein die Regel, dass „Denies“ stärker wirken als „Grants“.

   Ist ein Benutzer beispielsweise Mitglied in zwei Gruppen A und B, wobei für Gruppe A das Recht explizit erlaubt sei, für Gruppe B aber explizit verboten, so ist das Ergebnis der Rechteauswertung negativ. Somit kann der mögliche Zugriff auf Objekte durch Mitgliedschaften in Gruppen nicht nur größer werden, sondern kann sich auch reduzieren.

Dies ist der Grund dafür, dass …

- … die Verwendung von „Denies“ nur dort stattfinden soll, wo es unbedingt notwendig ist. In den meisten Fällen reicht es aus, mit den „Grant“- und „Not Set“-Einstellungen zu arbeiten.
- … Benutzer *nur entweder in „Users“ oder „Administrators“* Mitglied sein dürfen. Wäre ein Benutzer in beiden Gruppen Mitglied, würde einerseits zwar durch die Mitgliedschaft in „Administrators“ Zugriff auf besonders wichtige Verzeichnisse und Objekte im System gewährt werden. Andererseits aber verhindert die gleichzeitige Mitgliedschaft in „Users“, dass diese positiven Rechte zum Tragen kommen, da an einigen Stellen im System für „Users“ explizite „Denies“ eingetragen sind, die stärker wirken.

Wird kein passender Eintrag gefunden, ist das Ergebnis der Rechteüberprüfung per Definition negativ, d. h., der Zugriff wird nicht gewährt.

E.3.3 Bedienung

Um die gültige ACL für ein Objekt einzusehen, muss dieses im Explorer markiert werden. Anschließend öffnet ein Klick auf „`Rights`“ einen Popup-Dialog, welcher die momentan gültige ACL anzeigt. Die aktuellen Rechte können nur dann angesehen werden, wenn man auf das jeweilige Objekt das Recht *„ChangeRights“* besitzt, vgl. E.4.5. An dieser Stelle wird nochmals der Hinweis gegeben, für die WeLearn-Server im Browser eventuelle Popup-Blocker zu deaktivieren, da die Rechteverwaltung in einem separaten Fenster erfolgt.

Der Dialog in Abb. E-2 besagt, dass dieses Objekt keine eigene ACL definiert, sondern diese erbt. Die verwendete (also geerbte) ACL wird im strichliert-umrahmten Bereich dargestellt, wobei ein Häkchen besagt, dass dieses Recht explizit auf *„Grant“* gesetzt ist, ein Kreuzchen, dass das Recht explizit *„denied“* (verwehrt) ist, und ein leeres Feld, dass der Standard-Status *„not set“* gilt.

Um für ein bestimmtes Objekt eine eigene ACL zu definieren, muss die Rechtevererbung unterbrochen werden. Dazu muss auf den Button „`Set own rights`“ geklickt werden.

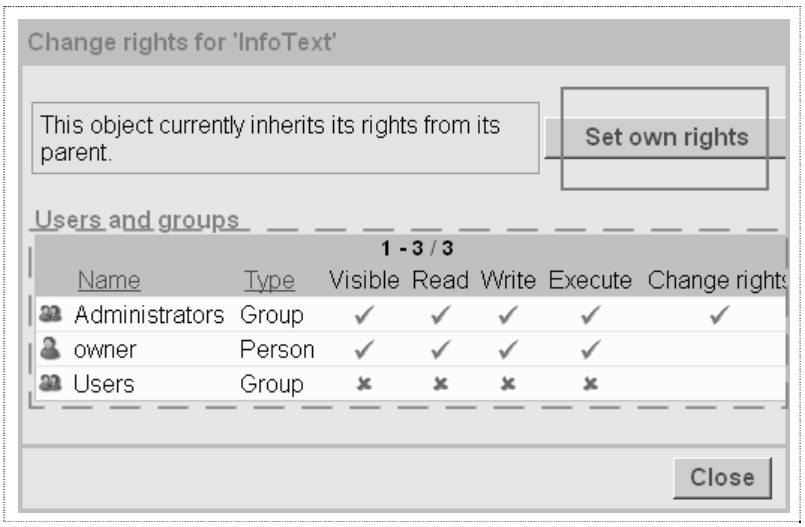

Abb. E-2 *Rechtedialog bei vererbten Rechten*

Nachdem die Vererbung unterbrochen und somit eine eigene ACL für das Objekt gesetzt wurde, sieht der Rechtedialog wie in Abb. E-3 aus.

Die vorher vererbten Rechte werden als Voreinstellung in die eigene ACL übernommen. Mittels Klick auf „Inherit rights from parent“ kann die Vererbung wieder eingeschaltet und die eigene ACL verworfen werden.

Die einzelnen Einträge für die Personen und Gruppen können jeweils in der linken Spalte markiert und anschließend gelöscht bzw. bearbeitet werden. In Abb. E-4 wurde der „Users“ Eintrag zum Bearbeiten ausgewählt. Man sieht, dass jedes der einzelnen Rechte auf einen der drei Zustände *„Deny“*, *„Not set“* bzw. *„Grant“* gesetzt werden kann. Zum Übernehmen der Einstellungen muss „Apply changes“ aufgerufen werden.

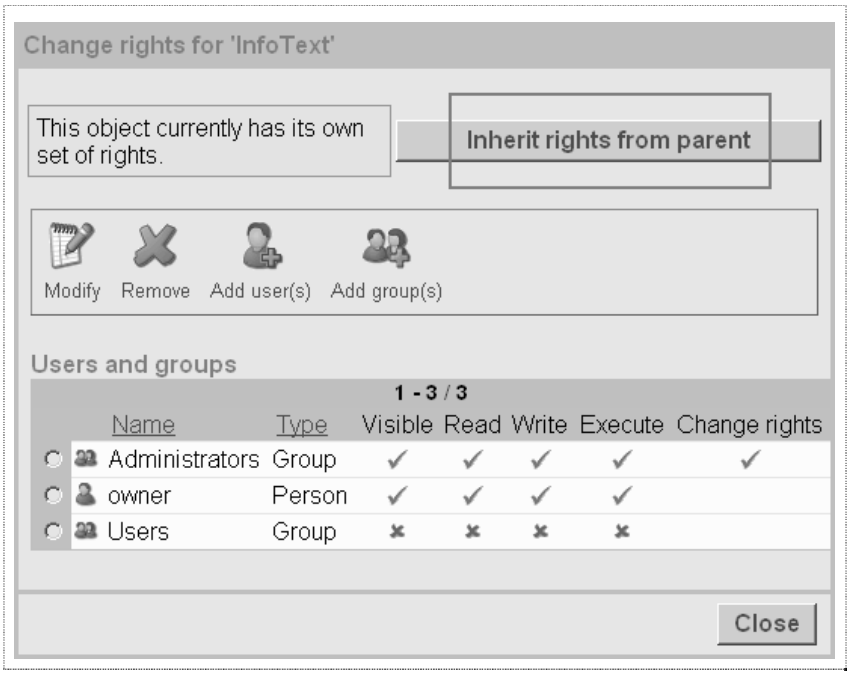

Abb. E-3 *Rechtedialog bei eigener ACL*

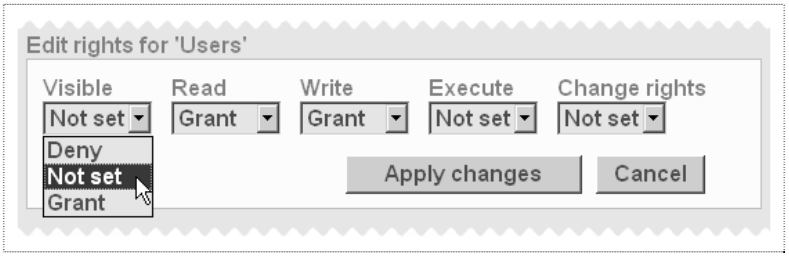

Abb. E-4 *Konfigurieren der Rechte für die Gruppe “Users”*

E.4 Arten von Rechten

In diesem Kapitel werden kurz die in WeLearn verfügbaren Rechte und deren Bedeutung besprochen.

E.4.1 Visible (Sichtbarkeit)

Über dieses Recht wird gesteuert, welche Benutzer im System das Objekt überhaupt angezeigt bzw. aufgelistet bekommen. Administratoren können über dieses Recht bspw. steuern, welche Benutzer welche Einträge im Top-Menü, in der Explorer- oder Kursansicht sehen.

Ist ein Objekt nicht sichtbar, wird es nicht angezeigt und kann weder ausgewählt noch aufgerufen werden. Somit ist das *„Visible“*-Recht überhaupt die Voraussetzung dafür, dass mit einem Objekt gearbeitet werden kann.

E.4.2 Read (Lesen)

Mittels *„Read“* kann gesteuert werden, für welche Benutzer der Inhalt von Objekten (zum Ansehen / Lesen) verfügbar ist. *„Read“*-Rechte sind auch die Voraussetzung, dass beispielsweise die Daten von aufgerufenen Foren, Messageboards, etc. angezeigt werden. *Read* soll in den meisten Fällen auf denselben Status wie das „Execute“-Recht gesetzt sein. Weiters wird das *Read*-Recht auch dazu verwendet, festzulegen, welche Objekte von Benutzern mittels der „Export“-Funktion aus dem System exportiert bzw. heruntergeladen werden können.

E.4.3 Write (Schreiben)

Für Containerobjekte wie Folder oder Menüs legt das *„Write“*-Recht fest, wer im Container neue Objekte anlegen darf. Bei anderen Objekttypen, wie Foren, MessageBoards, etc. wird mittels *„Write“* gesteuert, wer im Objekt neue Inhalte anlegen / schreiben darf. In einem Kalender bspw. legt das *Write*-Recht fest, welche Benutzer neue Termine eintragen dürfen. In einem MessageBoard wird *„Write“* dazu verwendet, um den Schreib-Zugriff auf das Board festzulegen.

E.4.4 Execute (Ausführen)

Das *Execute*-Recht muss in Zusammenhang mit „Read“ betrachtet werden und steuert den Zugriff auf ein Objekt. Für das Aufrufen („Anklicken“) eines Objektes im Top-Menü, in der Kursansicht oder im Explorer sind *Execute*-Rechte notwendig. Sind diese für einen Benutzer nicht gesetzt, wird ein Objekt nicht als Link dargestellt und kann somit vom Benutzer nicht geöffnet werden.

E.4.5 ChangeRights (Rechte einsehen und bearbeiten)

Um nicht jedem Benutzer, der Schreibzugriff auf ein Objekt hat, auch die Kontrolle über die ACL des Objektes zu geben, aber um andererseits nicht nur den Administratoren, sondern auch anderen „ausgewählten“ Benutzern die Bearbeitung von ACLs zu ermöglichen, wurde das Recht *„ChangeRights“* eingeführt. Über dieses kann detailliert gesteuert werden, wer die Rechte eines Objektes einsehen und verändern darf.

E.4.6 Edit (Bearbeiten)

Das Recht *„Edit“* ist nur bei einigen Objekten verfügbar. Diese sind Foren, MessageBoards sowie Kalender. *„Edit“* besagt bei diesen Objekten, ob ein Benutzer Beiträge verändern bzw. löschen darf. Ein Benutzer, der die Rolle eines Tutors für einen Kurs hat, könnte beispielsweise das *Edit*-Recht auf ein Forum haben, um etwaige Spam-Postings von Benutzern löschen zu können.

E.4.7 Attach (Anhänge posten)

Foren bieten als einzige Objekte in WeLearn die Möglichkeit, das *„Attach“*-Recht zu spezifizieren. Ist dieses Recht für einen einzelnen Benutzer oder für eine Gruppe gesetzt, kann in einem Forum zu einem eigenen Posting ein Anhang hinzugegeben, also eine lokale Datei hochladen werden.

E.5 Rechtevergabe in der Praxis

In der Praxis erwies es sich als sinnvoll und praktikabel, die einzelnen Rollen, die Benutzer in einem Kurs haben können, als individuelle Gruppen einzurichten. Somit könnten beispielsweise für einen Kurs *„Pflanzenkunde“ (PK)* folgende Gruppen im System eingerichtet werden:

- PK_Teilnehmer
- PK_Lehrer
- PK_Tutor

Das Anlegen des Kurses selbst wird von einem Administrator durchgeführt, indem im Ordner „/Courses“ ein neues Menü erstellt wird, welches „Pflanzenkunde“ genannt wird.

Die letzte Aufgabe des Administrators bei der Einrichtung eines neuen Kurses ist es, die Rechte auf den Kurs entsprechend zu setzen. Es wird bei dem Kursobjekt die Vererbung gebrochen, der Gruppe „PK_Lehrer“ werden alle Rechte gegeben (auch *„ChangeRights“*), die Gruppe „PK_Teilnehmer“ erhält *„Read“*-, *„Write“*- und *„Execute“*-Rechte. Je nachdem, ob der Kurs für alle Benutzer des Systems sichtbar sein soll, müssen noch die Rechte für die Gruppe „Users“ geändert werden. Kurse sind standardmäßig für alle Benutzer des Systems zwar sichtbar, ebenso die Kursinhalte, sie können aber nicht abgerufen und angeklickt werden.

Dadurch, dass die Gruppe PK_Lehrer auch *„ChangeRights“* erhält, hat/haben der/die Kursleiter volle Kontrolle über den Inhalt und die Rechte des Kurses. Es können somit verschiedenste Lernszenarien nachgebildet werden. Einzelarbeiten, öffentliche bzw. auch private Gruppenbereiche, etc. können realisiert werden, da der Kursleiter innerhalb des Kurses durch die Rechtevergabe wiederum spezielle Bereiche erstellen kann. Die Rechte für die Tutoren können gezielt auf spezielle Objekte von den/vom Lehrenden selbst vergeben werden.

Allgemein ist die Administration der Rechte meist mit wenig Aufwand verbunden, da die Rechte nur einmal beim Kursordner gesetzt werden müssen und von allen im Kurs befindlichen Objekten wie Lernmaterial, Kalender, Ordner, etc., übernommen werden.

Änderungen in den Rechten müssen daher nur an wenigen Stellen, wie z. B. bei Foren, Abgabeordnern, o. ä. explizit vorgenommen werden. Meist ist dies dann notwendig, wenn für Teilnehmer eines Kurses beispielsweise auch Schreibrechte eingerichtet werden sollen.