

Die OLAT-Erweiterung XMAN für Prüfungsanmeldungen

Daniel Gerber*
Institut für Informatik
Universität Leipzig

Hans-Gert Gräbe†
Institut für Informatik
Universität Leipzig

Zusammenfassung

Wir stellen eine OLAT-Erweiterung vor, mit der das operative Geschäft zur Verwaltung von Prüfungen auf der Ebene einer Fakultät – dem „Owner“ der entsprechenden Prozesse – von der Anmeldung bis zur Bekanntgabe und Überführung der Noten in zentrale Datenerfassungssysteme webgestützt abgewickelt werden kann. Die abgebildeten Prozesscharakteristika orientieren sich (natürlich) an den entsprechenden Festlegungen in unserer Fakultät, dürften sich aber auf andere Einrichtungen weitgehend übertragen lassen.

Die Erweiterung wurde im Rahmen der studentischen Ausbildung im Bachelorstudiengang Informatik über zwei Jahre bis zur heutigen Reife entwickelt.

1 Einleitung

Im Rahmen der studentischen Ausbildung im Fach Softwaretechnik im Bachelorstudiengang Informatik werden am Institut für Informatik der Universität Leipzig bereits über mehrere Jahre verschiedene Erweiterungen des LMS OLAT [2] konzipiert und prototypisch entwickelt. Es zeigt sich, dass an diesem Beispiel praktische Aspekte komplexer softwaretechnischer Fragen wie die Konzipierung einer Architektur oder der Einsatz von Frameworks fortgeschrittenen Bachelorstudenten gut vermittelt werden können, die sich auch bis zu Qualifizierungsarbeiten fortführen lassen. Hier soll über ein solches Projekt berichtet werden, das in den letzten zwei Jahren im BIS-OLAT-Team vorangetrieben und mit der Bachelorarbeit [1] von Herrn Gerber zu einem ersten Abschluss geführt wurde.

Ziel des Projekts war die Ablösung einer an der Fakultät eingesetzten Eigenentwicklung auf PHP-Basis zur Prüfungsanmeldung, die verschiedene Sicherheitsmängel und Datenflussbrüche aufwies und den Prozess der Prüfungsabwicklung in seiner vollen Dimension nur rudimentär unterstützte. Die neue Software sollte als Erweiterung von OLAT konzipiert werden und den Prozess der Prüfungsabwicklung vom Ansetzen der Prüfungstermine über das Einschreiben der Teilnehmer bis zur Bekanntgabe der Noten und deren digitale Übermittlung an das Prüfungsamt auf einheitliche Weise unterstützen.

OLAT bietet mit seinem Rechte- und Rollenkonzept sowie verschiedenen Erweiterungspunkten sehr gute Voraussetzungen, eine solches Projekt relativ selbstständig zu entwickeln und als OLAT-Erweiterung auch zu betreiben. Es war allerdings zu beachten, dass auf Grund

*daniel.gerber@me.com

†graebe@informatik.uni-leipzig.de

der zentralen Betreiberszenarien der wichtigsten OLAT-Anwender Prozessketten in der operativen Verantwortung einzelner Einrichtungen im OLAT-Design kaum berücksichtigt sind, hier also konzeptionelles Neuland betreten wurde.

Die zu entwickelnde Erweiterung stand also nicht nur vor technischen Herausforderungen wie der Erweiterung des OLAT-Rollenkonzepts um eine weitere Rolle *Prüfungsamt* oder der Einbindung einer neuen Lernressource *Prüfungen* – für beides gab es in den verwendeten OLAT-Versionen 6.0 und 6.1 keine Erweiterungspunkte –, sondern auch vor der spannenden Frage, ob sich diese neuartigen prozessualen Aspekte auf lokaler Ebene überhaupt adäquat im OLAT-Framework abbilden lassen. Mit unseren Ergebnissen können wir OLAT in dieser Frage ein sehr gutes Zeugnis ausstellen.

Die Erweiterung XMAN wurde in einer ersten prototypischen Version im Software-technik-Praktikum 2007 entwickelt und im darauffolgenden Semester konsolidiert. Nach entsprechenden Absprachen an der Fakultät kam die so entwickelte Erweiterung in einer lokalen Produktivinstanz zur Prüfungsanmeldung für ausgewählte Veranstaltungen zum Einsatz und hat sich dort inzwischen über mehrere Semester bewährt. Der Prüfungsausschuss forderte von Anfang an, dass diese Instanz aus datenschutzrechtlichen Gründen streng von den operativen Informationen zur Begleitung von Lehrveranstaltungen – dem eigentlichen Einsatzgebiet von OLAT – zu trennen sei, so dass wir zwei lokale OLAT-Produktivinstanzen betreiben, das *BIS-OLAT-Portal* zur Begleitung von Lehrveranstaltungen und die *OLAT-PA-Instanz* mit der XMAN-Erweiterung zur Prüfungsanmeldung.

Eine solche Konstellation ist aus mehreren Perspektiven interessant. Zum einen liegen damit erste Erfahrungen auch in der Interaktion verschiedener OLAT-Instanzen vor, was unter dem Portal-Fokus von den OLAT-Entwicklern bisher generell nicht betrachtet wurde. Zum anderen wird deutlich, dass mit einer solchen Auftrennung in verschiedene Instanzen nicht nur eine Reihe datenschutzrechtlicher Fragen präziser adressiert werden kann, sondern sich auch lokale Besonderheiten besser berücksichtigen lassen.

Unser Szenario geht zwar davon aus, dass diese lokalen Instanzen auch vor Ort betrieben werden, dies ist jedoch für Web-Applikationen keine wesentliche Einschränkung, wenn zwischem dem technischen Betrieb und der Administration der Inhalte unterschieden wird. Der technische Betrieb der verschiedenen Instanzen könnte durchaus an einen externen Betreiber wie etwa die BPS ausgelagert werden, der zugleich neue technische Anforderungen sammelt, in den OLAT-Inkubationsprozess einbringt, sich an deren Realisierung beteiligt und im Übrigen die Übersicht behält, welche der von ihm betriebenen Instanzen wie konfiguriert sind. Die hierbei einzusetzenden Techniken des Produktlinien-Managements sind in der Software-Technik ausreichend weit entwickelt, so dass sich auch hier eine Verzahnung mit der studentischen Ausbildung anbietet, dann allerdings auf dem Niveau der Master-Ausbildung.

2 Ableitung der prinzipiellen Aufgabenstellung

Ausgangspunkt der Entwicklung war der von unserem Prüfungsausschuss am 9.1.2007 beschlossene Workflow für das Prüfungsmanagement sowie die Elemente der Umsetzung desselben in der bestehenden fakultätsinternen PHP-basierten Lösung.

Die bisherigen Erfahrungen zeigen, dass in etwa 90 % der Anmeldungsfälle die Prüfungskandidaten gutwillig kooperieren. Dies betrifft vor allem „normale“ Prüfungsanmeldungs-fälle, während im Bereich der Wiederholungsprüfungen die Kooperationsrate deutlich gerin-

ger ist. So reagieren einige Studenten trotz wiederholter E-Mail-Aufforderung zur Einschreibung nicht. Es handelt sich dabei meist um Studenten, die im Prüfungsamt der Fakultät schon bekannt sind, weil sie nur sehr unregelmäßig studieren oder sich mit dem Gedanken tragen, das Studium ganz aufzugeben. In klaren Fällen kann im PA-Modul XMAN diesen Studenten nach Ablauf der Anmeldefrist durch den Prüfer oder andere Verantwortliche zwar auch digital ein Termin zugewiesen werden, aus den bekannten allgemeinjuristischen Gründen erfordert eine gerichtsfeste Mitteilung des Prüfungstermins einen eingeschriebenen Brief oder offiziellen Aushang, in jedem Fall aber eine direkte Intervention des Prüfungsamts. Eine solche Intervention kann IT-gestützt vorbereitet und begleitet, nicht aber vermieden werden.

Vollkommen unklar ist die Situation bei zweiten Wiederholungsprüfungen, die einen genehmigten Antrag erfordern. In der Regel wird dieser von den Kandidaten sehr schleppend und nicht im aktuellen Prüfungszeitraum gestellt, so dass sich hier „Legacy“-Fälle auf natürliche Weise ergeben. Auch restriktive Festlegungen des Prüfungsausschusses haben hier keine grundsätzliche Wende herbeigeführt. Noch unübersichtlicher wird die Situation, wenn verschiedene Aspekte von Studiengangwechsel, krankheitsbedingte Pausen, Urlaubs- und Auslandssemester usw. zu berücksichtigen sind.

Dies alles zusammen legt nahe, die Prüfungsanmeldung als **IT-Unterstützung** für einzelne Prüfungen auszulegen, mit der gutwillig studierenden Kandidaten eine bequeme und sichere Form der Handhabung des Prüfungsmanagements an die Hand gegeben wird, mit Wiedervorlagepunkten in Problemfällen, wobei auf klare und nachvollziehbare Weise der „Staffelstab“ an das Prüfungsamt oder andere für die Prüfung Verantwortliche zurückgegeben wird. Dies entspricht auch den flexiblen Anforderungen an die verschiedenen Akteure bzgl. der im PA-Workflow vom 9.1.2007 festgelegten Verteilung von Entscheidungs- und Interventionsverantwortlichkeiten. Auf Grund der komplexen Interdependenzen zwischen Prüfung und Nachprüfung ist es sinnvoll, beide auch in derselben Prüfungsperiode zu trennen und als eigenständige Anmeldeverfahren zu behandeln.

Die PA-Lösung ist deshalb als **IT-Unterstützung für das operative Geschäft von der Anmeldung bis zur Bekanntgabe der Ergebnisse einer Prüfung und deren Einarbeitung in zentrale Datenspeicher** ausgelegt. Es wird dazu begrifflich unterschieden zwischen

1. der **PA-Lösung** – das Software-Artefakt als Erweiterung von OLAT,
2. dem **PA-Portal** – eine durch die Verantwortlichen konfigurierte und betriebene Instanz der PA-Lösung,
3. dem **PA-Prozess** als konfigurierbarem Template innerhalb des PA-Portals, welches die IT-Unterstützung des spezifizierten Prüfungsanmeldungs-Workflows umsetzt, sowie
4. konkreten PA-Prozessinstanzen und den zugehörigen Daten tragenden Softwareobjekten, den **PA-Instanzen**.

Der weiter unten dargestellte detaillierte PA-Workflow geht davon aus, dass

- eine die Prüfungen koordinierende Instanz (Prüfungsamt) existiert,
- ein für die aktuelle Prüfung Verantwortlicher (P-V) festgelegt ist,
- alle potenziellen Kandidaten am PA-Portal registriert sind und dort über eine *elektronische studentische Akte* (ESA) verfügen, über die weitere Informationen beigezogen werden können.

Diese ESA ist als *Ergänzung*, nicht als Substitut der bisherigen studentischen Handakten

gedacht und dient primär der *Verwaltung operativer Informationen* wie Krankschreibungen usw., deren Papierform auch noch einmal getrennt zu archivieren ist. Um die Papierform wird man in absehbarer Zeit aus rechtlichen Gründen nicht herkommen.

Eine PA-Instanz enthält strukturierte Informationen über

- Online-Anmeldezeitraum, der so zu wählen ist, dass nach dessen Ablauf weitere Zwangseintragungen rechtswirksam möglich sind,
- Zulassungsbedingungen und rechtliche Konsequenzen der Anmeldung (nach Vorlage, welche durch die P-V modifiziert werden kann) – auf diese wird im Dialog zur Prüfungseinschreibung noch einmal hingewiesen,
- Gruppe der anmeldeberechtigten Kandidaten (A-K, realisierbar als OLAT-Gruppe, derzeit noch nicht umgesetzt)
- Ort und Termin (schriftlich) bzw. Terminliste (mündlich) der Prüfung. Es sind die Modi „ein Termin, viele Kandidaten“ (schriftlich) und „viele Termine, jeweils ein Kandidat“ (mündlich) vorgesehen.
- Standardisierte E-Mail-Benachrichtigung der A-K an relevanten Punkten (An- und Abmeldung, Notenbekanntgabe) des PA-Workflows.

3 Behandlung studentischer Identitäten am PA-Portal

Der folgende Mechanismus der Feststellung der studentischen Identität (genauer – der Zuordnung realer Personen zu digitalen Identitäten) geht davon aus, dass

1. die praktische Handhabung für alle Seiten ausreichend bequem sein soll,
2. das IT-System unterstützend eingesetzt ist und im Streitfall nicht als unumstößlicher rechtsgültiger Beweis herangezogen werden kann,
3. eine hohe Plausibilität der Identität vorliegt, wenn mehrere Aktionen unter dieser Identität zweifelsfrei demselben realen Nutzer zugeordnet werden können (Vertrauensgrundsatz).

Aktuell nutzen wir die zentrale LDAP-Authentifizierung an der Universität Leipzig für die Zuordnung von digitalen Identitäten zu realweltlichen Personen und erfassen weitere prüfungsrelevante Informationen im Zuge der *Validierung der elektronischen Studentenakte*.

Den einzelnen studentischen Identitäten sind Informationen aus zwei Bereichen zugeordnet,

- (a) einem durch den Studenten privat verwalteten Bereich (OLAT-Standard, steht nach Registrierung am Portal sofort zur Verfügung) und
- (b) einem administrativ verwalteten Bereich, der – vom Studenten zu beantragenden – elektronischen Studentenakte (ESA).

Bereich (a) wird ausschließlich durch die Studenten selbst verwaltet, die auch Sichtbarkeitsregeln für private Informationen nach dem OLAT-Standard selbst einstellen können. Bereich (b) wird vom Prüfungsamt verwaltet und kann von der Rolle *Prüfungsamt* und vom jeweiligen Studenten eingesehen werden.

Authentifizierung, Autorisierung und Kommunikation erfolgen wie in OLAT über (a). Insbesondere können sich so auch Studenten anderer Fakultäten am Portal registrieren, wenn sie bei uns Prüfungen absolvieren.

Prüfungsrelevante Informationen werden an die Studserv-Adresse verschickt. Dabei wird zwischen *einfacher E-Mail* und *amtlicher E-Mail* unterschieden. Die Versendung einfacher E-Mails kann nur aus dem Log-File nachvollzogen werden, amtliche E-Mails werden darüber hinaus in der ESA vermerkt (noch nicht implementiert).

Die ESA enthält die folgenden amtlichen Informationen:

- (1) Name, Vorname
- (2) Studserv-Account und Matrikelnummer
- (3) Studiengang (aus einer vom Prüfungsamt verwalteten Liste)
- (4) Informationen zu bisherigen Prüfungsergebnissen
- (5) Informationen über Krankschreibungen
- (6) Freitextbereich für weitere Notizen autorisierter Personen, die zur Abwicklung des operativen Geschäfts erforderlich sind.

(1) – (3) wird von den Studenten im Zuge der **Beantragung der ESA** (Link im Home-Bereich des Studenten) eingetragen und von der Rolle *Prüfungsamt* validiert. Änderungen können von den Studenten auf demselben Weg vorgenommen werden und sind ebenfalls von der Rolle *Prüfungsamt* zu validieren. Nur Studenten mit validierter ESA können sich für Prüfungen anmelden.

Die ESA ist so konzipiert, dass über einen genau ausgeprägten Erweiterungspunkt später weitere strukturierte Informationen und Funktionalitäten hinzugefügt und von neu definierbaren Rollen modifiziert werden können.

Der Student hat lesenden Zugriff auf beide Bereiche und kann sich so jederzeit über die zur eigenen Person gespeicherten Informationen ins Bild setzen. In die ESA dürfen nur autorisierte Personen schreiben. Weiterhin werden dort im PA-Workflow vereinbarte Informationen automatisiert abgelegt.

In der aktuellen Version wird die Verbindung von Student und ESA durch das Zusammenfallen von Name, Vorname, Studserv-Account und Matrikelnummer mit der Eintragung in amtlichen Listen des Prüfungsamts durch die Rolle *Prüfungsamt* manuell geprüft und validiert. Dazu werden dem Prüfungsamt die aktuellen Anmeldungen in Form einer Checkliste angeboten, die regelmäßig abzuarbeiten ist. Bei Übernahme der Daten erhalten die Studenten eine automatisierte einfache Bestätigungs-E-Mail. Bei Unstimmigkeiten kann das Prüfungsamt über das personenbezogene OLAT-Kontaktformular, welches standardmäßig über einen Link in jeder Zeile der Liste aktivierbar ist, mit dem betroffenen Studenten per E-Mail eine Klärung herbeiführen.

Diese Informationen liegen dem Prüfungsamt für die bei uns immatrikulierten Studenten vor und müssen von Fakultäten, zu denen Fächerkooperationsvereinbarungen bestehen, in zu vereinbarenden Form abgefordert werden, die eine digitale Verarbeitung möglich macht. Durch entsprechende Werkzeuge ist das digitale Einlesen solcher Informationen in einer weiteren Ausbaustufe vorgesehen.

Da es sich um einen für die Portallebensdauer einmaligen Identifizierungsvorgang handelt, ist der damit verbundene Aufwand zu rechtfertigen. In einer weiteren Ausbaustufe ist eine Validierung der Angaben auch über eine universitätseinheitliche zentrale Datenquelle denkbar. Dafür fehlen derzeit allerdings die technischen und rechtlichen Voraussetzungen.

4 Detaillierter PA-Workflow

Vorbedingung: Alle anmeldeberechtigten Studenten sind am Portal registriert und haben eine ESA.

1. Der Prüfungsverantwortliche (P-V) legt eine neue PA-Instanz an. Damit ist zugleich die Rolle *Besitzer der PA-Instanz* besetzt.

Der P-V kann die direkte Eintragung der anmeldeberechtigten Prüfungskandidaten (A-K) zur Prüfung freischalten, aber auch einen *Vormerkmodus* aktivieren, wenn die Prüfungszulassung von weiteren Voraussetzungen abhängig ist, etwa dem Vorliegen eines Übungs- oder Seminarscheins.

Derzeit können sich alle am Portal registrierten Personen mit validierter ESA in jede Prüfung eintragen. In einer späteren Ausbaustufe soll die Gruppe der A-K auch von vornherein eingeschränkt werden können. Dies lässt sich über die Steuerung von Zugriffsrechten durch die OLAT-Gruppenfunktion realisieren. Der P-V soll diese Funktionalität aktivieren und die Gruppe der A-K über die Standard-OLAT-Funktionen für Gruppen anlegen können.

2. Die PA-Instanz wird im Zuge des OLAT-üblichen Publikationsprozesses von der Koordination (durch den P-V, bei Klausuren nach Rücksprache mit dem Prüfungsamt) freigegeben.
3. Die A-K nehmen die Anmeldeoption wahr. Der P-V überprüft zeitnah die Erfüllung der Zulassungsvoraussetzungen der vorgemerkten Kandidaten und gibt ggf. die Anmeldung für die Prüfung frei.

In einer späteren Ausbaustufe kann der A-K im Kommentarfeld angeben, wofür die Prüfung angerechnet werden soll und um den wievielten Versuch es sich handelt.

An- und Abmeldungen sowie Freigabe bei vorgemerkter Teilnahme werden per *einfacher E-Mail* bestätigt. In der E-Mail wird über die Rechtsfolgen per Standard-Textbaustein umfassend informiert.

In einer späteren Ausbaustufe kann dieser Standard-E-Mail-Text im Zuge der Erzeugung der PA-Instanz durch den P-V modifiziert werden.

4. Nach Abschluss der Online-Anmeldung sind weitere Termine an Kandidaten zu vergeben, welche die Prüfung absolvieren müssen, sich aber nicht rechtzeitig online angemeldet haben.

Dazu werden die Kandidaten direkt durch den P-V oder über das Prüfungsamt kontaktiert, rechtsverbindlich ein Prüfungstermin vereinbart oder bekanntgegeben und die Prüfungsanmeldung im Portal nachgetragen. Der Student erhält dazu eine *amtliche E-Mail*.

Die Möglichkeit zum Nachtragen ist für den P-V bis zum Ende der Prüfung möglich, um Studenten zwecks Notenmitteilung nachtragen zu können, die über einen anderen als den elektronischen Weg eine Prüfungszulassung erlangt haben.

5. Austragungen aus der Prüfung erfolgen im Ausschlusszeitraum über den P-V oder das Prüfungsamt. Der Student erhält dazu eine *amtliche E-Mail*.

6. Der P-V hat sowohl während der Online-Anmeldung als auch während des Ausschlusszeitraums die Möglichkeit, mit einzelnen Kandidaten über weitere spezielle rechtliche Konsequenzen (insb. wenn diese zur Anmeldung verpflichtet sind) über die Kontaktfunktion von OLAT zu kommunizieren.

Dabei getroffene Festlegungen und Vereinbarungen werden durch den P-V per *amtlicher E-Mail* bestätigt.

7. Der P-V kann sich eine Liste der Prüfungsteilnehmer im Excel-Format herunterladen, um daraus abgeleitete Dokumente (Prüfungsprotokolle, Unterschriftenlisten) herzustellen bzw. herstellen zu lassen.
8. Nach dem Prüfungstermin werden nicht angetretene Teilnehmer dem Prüfungsamt gemeldet, um deren Status (krank, Note 5 wegen Fernbleibens) in Erfahrung zu bringen.
In einer späteren Ausbaustufe soll dies über eine direkte Portalfunktion möglich sein.
9. Nach Abschluss der Prüfung werden die Ergebnisse durch den P-V eingetragen und freigegeben. Eine einfache E-Mail informiert die Prüflinge darüber sowie über bestehende Einsicht- und Einspruchsmöglichkeiten. Prüflinge sehen im PA-Portal nur ihr eigenes Ergebnis.
10. Das Prüfungsamt und die P-V arbeiten Änderungen durch Einsprüche, Krankenschein usw. ein.
11. Aus der PA-Instanz werden Ergebnisliste und Informationen über Nachprüfungen als digital weiter verarbeitbare Datensätze durch das Prüfungsamt extrahiert.
12. Die PA-Instanz wird geschlossen und digital archiviert.
Wegen der Verbindung von PA-Instanzen und ESA-Einträgen kann eine PA-Instanz derzeit nicht gelöscht werden.

Zur Entlastung des Prüfungsamts ist dabei der **Aufbau eines Systems von Verantwortlichen** sinnvoll, wie es praktisch schon an einzelnen Abteilungen besteht. Diese Verantwortlichen sind der manuelle operative Interventionspunkt für die ihnen zugeordneten Prüfungen. Dazu lassen sich im PA-Modul XMAN Prüfungen zu Modulen zusammenfassen.

5 Die technische Lösung

XMAN besteht aus zwei großen Teilsystemen. Das ist zum einen die Komponente *Prüfung*, welche aus einer Menge von Unterkomponenten wie zum Beispiel Modulen, Terminen und den Prüfungsinstanzen selbst besteht, und zum anderen die Komponente *Elektronische Studentenakte*, die wiederum aus Unterkomponenten wie einer Menge an persönlichen Informationen, Kommentaren und Krankenscheinen zusammengesetzt ist. Zu diesen persönlichen Informationen zählen der Vor- und Nachname, die Matrikelnummer, der Studiengang sowie die Studserv-Emailadresse des jeweiligen Studenten. Die beiden Teilsysteme können theoretisch getrennt voneinander existieren, das heißt, es existieren keinerlei Abhängigkeiten zwischen ihnen. Allerdings wäre der Funktionsumfang beider Komponenten stark eingeschränkt, sollten sie einzeln zum Einsatz kommen. Beispielsweise hätte das Prüfungsamt

keinen direkten Überblick über die Noten eines Studenten oder es wäre lediglich möglich, über Studenten Kommentare und Krankenscheine zu erfassen anstatt deren Prüfungsergebnisse zu verwalten. Die Kopplung der beiden Systeme an OLAT ist hingegen sehr ähnlich. Sowohl die Prüfung als auch die elektronische Studentenakte nutzen bestehende Erweiterungspunkte und gliedern sich somit nahtlos in OLAT ein.

5.1 Module und Studiengänge

Wie bereits erwähnt, können Prüfungen zu Modulen zusammengefasst werden. Als Modul wird in diesem Kontext ein zeitlich und inhaltlich abgeschlossener Verbund von Lehrveranstaltungen mit einem ähnlichen inhaltlichen und thematischen Schwerpunkt verstanden. Diese Zuordnung von Prüfung und Modul geschieht in der aktuellsten Version direkt beim Erzeugen einer Prüfungsinstanz. Das dem Examen zugeordnete Modul muss dabei mindestens die Attribute Namen, Beschreibung und einen für alle Prüfungen dieses Moduls verantwortlichen Mitarbeiter des Lehrstuhls besitzen. Optional ist hier zusätzlich noch die Angabe einer dem offiziellen Standard entsprechenden Modulnummer möglich. Des Weiteren benötigt das Prüfungsamt, neben dem abzulegenden Modul, Informationen über den Studiengang des Prüflings. Da OLAT für die Erfassung des Studienganges lediglich ein Freitextfeld zur Verfügung stellt, musste ein alternativer Lösungsansatz gefunden werden, um eine eindeutige Zuordnung zu gewährleisten. Bei der Recherche zur Lösung dieses Problems wurde klar, dass der eleganteste und flexibelste Weg über den bereits existierenden Erweiterungspunkt *UserPropertyHandler* führt. Hinter diesem Begriff verbirgt sich ein Managerobjekt, welches über bestimmte Methoden die Eigenschaften, also auch den Studiengang eines Nutzers, in Formularen darstellen und manipulieren kann, aber keine tatsächlichen Nutzerdaten enthält. Außerdem ist anzumerken, dass der *UserPropertyHandler* ein speziell für diesen Zweck bestimmtes Formularelement, beispielsweise eine Drop Down Box oder ein Check Box Feld, sowie eine Beschreibung des Spaltenkopfes für die Darstellung des Nutzerattributs in Tabellenform liefert. Nach dessen Implementierung und dem Hinzufügen der Beandefinition in die *olat.userconfig.xml*-Datei kann die neue Nutzereigenschaft in allen Formularen direkt eingebunden werden, ohne sich an diesen Stellen um die Darstellung oder die Manipulation dieser Informationen kümmern zu müssen.

Um nun auch eine Auswahl an Studiengängen anbieten zu können, wurde im Prüfungsamtbereich eine Möglichkeit zur Erzeugung und Umbenennung jener Daten zur Verfügung gestellt. Diese Editierfunktion beschränkt sich hier allerdings auf die vom jeweiligen Mitarbeiter eingestellte Standardsprache. Da OLAT diverse Funktionen zur Mehrsprachigkeit anbietet, können die Bezeichnungen der Studiengänge direkt über das zur Laufzeit ausführbare Übersetzungsprogramm in alle angebotenen Sprachen übersetzt werden.

5.2 Prüfungen

Die Hauptbestandteile von OLAT sind Lernressourcen. Der Begriff Lernressource wird dabei synonym für die Anlage aller verfügbaren Lerninhalte, zu erreichen über den Reiter '*Lernressourcen*', sowie für die an dieser Stelle einzeln abgelegten Lerninhalte verwendet. Ferner werden verschiedene Typen, wie etwa Kurse, Tests oder Wikis unterschieden. Insbesondere ist die Lernressource *Kurs* in der Lage, beliebig viele Kursbausteine, wie beispielsweise ein Forum zu beinhalten und damit fähig, Vorlesungen, Seminare oder Praktika abzubilden. Somit würde es nahe liegen, eine Instanz einer Prüfung als Kursbaustein eines vorhandenen

Kurses anzulegen und somit eine direkte Verbindung zwischen Lehre und Leistungsmessung zu schaffen. Allerdings wird dieser Zusammenschluss durch die Grundvoraussetzung des Prüfungsausschusses zur Trennung von Lehrveranstaltungen und Prüfungen, zumindest an der Fakultät für Informatik¹, verhindert. Aus dieser Voraussetzung ergibt sich als einzige akzeptable Lösung eine separate OLAT-Instanz, die in der Lage ist, Prüfungen als Lernressourcen zu interpretieren. Dies hat aber auf technischer Seite den Nachteil, dass OLAT für eine solche Form der Adaption keinen Erweiterungspunkt zur Verfügung stellt und – wie bereits das Hinzufügen einer neuen Rolle – Veränderungen am OLAT Quellcode erforderlich macht.

Für die folgende detaillierte Beschreibung der notwendigen Veränderungen werden vorab einige Begriffe aus der OLAT-Architektur kurz rekapituliert:

RepositoryEntry:

Ein *RepositoryEntry* ist eine, wie der Name bereits vermuten lässt, Repräsentation eines Eintrags in der Lerninhaltsablage. Dabei kapselt er nicht etwa die eigentlichen Daten einer Lernressource, sondern vielmehr deren Metainformationen wie den Autor, das Erstellungsdatum oder die Zugriffsrechte.

RepositoryMainController:

Der *RepositoryMainController* ist dafür zuständig, Lernressourcen eines vorgegebenen Typs zu erzeugen und sie ebenfalls wieder zu löschen. Er dient außerdem dazu, bestehende Lernressourcen zu importieren, einen Überblick über existierende Lernressourcen zu geben und eine dem Typ der Lernressource entsprechende Suche bereitzustellen.

RepositoryHandler:

Der *RepositoryHandler* ist ein Managerobjekt, das dafür zuständig ist, die zu einer bestimmten Lernressource gehörenden Controller für das Hinzufügen, Editieren, Starten und Löschen bereitzustellen. Außerdem dient er dazu, zu einer Lernressource gehörende Metainformationen, wie beispielsweise die Möglichkeit zum Download, zu liefern.

Wie in Abbildung 1 zu sehen ist, musste ein neuer *RepositoryHandler* für die Lernressource beziehungsweise den *RepositoryEntry* Prüfung angelegt werden. Dieser muss, um im gesamten System zur Verfügung zu stehen, in der *RepositoryHandlerFactory* registriert werden. Die eigentliche Implementierung *ExamHandler* der Schnittstelle *RepositoryHandler* stellt nun die für die Erzeugung, das Editieren und Starten der Prüfung notwendigen Controller zur Verfügung. Wird nun beispielsweise im *RepositoryMainController* der Befehl zum Erstellen einer neuen Prüfungsinstanz empfangen, leitet dieser den Aufruf an den *RepositoryAddController* weiter, der den entsprechenden *IAddController*, also in diesem Fall den *ExamCreateController* instanziiert. Außerdem existiert ein ähnliches, aber hier nicht weiter beschriebenes Verfahren zur Erzeugung der Instanzen der Controller für den Editierbeziehungsweise Startvorgang.

¹Für den Fall, dass diese strikte Trennung an einem anderen Einsatzort nicht bestehen muss, wurde die Möglichkeit zur Integration von Prüfungen in Kurse als Kursbaustein ebenfalls implementiert.

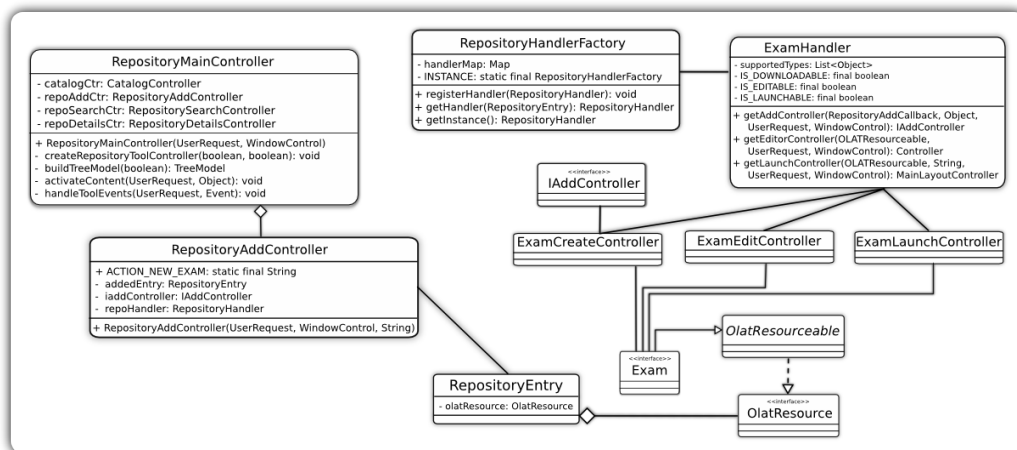


Abbildung 1: Prüfung als Lernressource

Struktur

Die Prüfung wurde bis jetzt lediglich in einem globalen Kontext betrachtet. Das heißt, es wurde gezeigt, auf welche Art und Weise die Prüfung in OLAT integriert wurde sowie welche anderen Komponenten zur Prüfungsverwaltung benötigt werden. Nun soll näher auf den eigentlichen Aufbau einer solchen Prüfung eingegangen werden. In diesem Abschnitt werden dabei ausschließlich die Modell- beziehungsweise die entsprechenden Managerklassen betrachtet, welche im Model-View-Controller Design Pattern strikt von den zur Anzeige und Steuerung benötigten Bestandteilen getrennt sind. Wie auch in Abbildung 2 zu sehen ist, haben sich zwei größere Hauptklassen, *ExamImpl* und *ProtocollImpl* sowie einige kleinere Hilfsklassen herausgebildet. Dabei repräsentiert eine Instanz der Klasse *ExamImpl* eine real von einem Prüfungsautor angelegte Prüfung. Die Unterscheidung zwischen schriftlichen und mündlichen Prüfungen geschieht hier über den booleschen Wert *isOral*. In einem objektorientierten Modell könnte man davon ausgehen, dass für diese zwei verschiedenen Prüfungsarten auch zwei verschiedene Implementierungen angeboten werden. Tatsächlich wurde dieser Ansatz auch zu Anfang der Entwicklungsarbeiten verfolgt, aber nach einiger Zeit aus Gründen des nicht gerechtfertigten Zusatzaufwands wieder verworfen. Dennoch könnte man, um der Objektorientierung besser zu genügen, die Klasse *ExamImpl* als abstrakt definieren und dort alle vorhandenen Methoden implementieren, bis auf die abstrakten Methoden *getAppointments()* und *setAppointments(List<Appointment> appointments)*. Nun könnten zwei zusätzliche Subklassen *OralExam* und *WrittenExam* nach den gegebenen Anforderungen erzeugt werden, die nur diese beiden Methoden implementieren. Jedem vorhandenen Termin wird genau eine Prüfung zugeordnet, das heißt im Umkehrschluss, dass einer schriftlichen Prüfung ebenfalls genau ein Termin zugeordnet ist (1:1 Beziehung). Einer mündlichen Prüfung hingegen wird eine Menge von Terminen zugewiesen (1:N Beziehung). Es wäre hier ebenfalls möglich gewesen, dass das Examen eine Liste aller sich selbst betreffenden Termine verwaltet, was aber eine zusätzliche Verknüpfungsrelation auf der Datenbankebene für diese Liste der mündlichen Prüfungen erfordert hätte. Die zweite große Komponente im Prüfungsmodul bildet die Klasse *ProtocollImpl*. Sie repräsentiert das eher bei mündlichen Prüfungen verwendete, in Papierform vorliegende Prüfungsprotokoll. Die Klasse wird in dem Moment instanziiert, in dem sich ein Student, hier durch die *Identity*

repräsentiert, zu einem bestimmten Termin einer Prüfung anmeldet. Nachdem die Prüfung und deren Korrektur abgeschlossen ist, kann der verantwortliche Prüfer etwaige Kommentare hinzufügen, bestimmten Studenten amtliche Emails senden und die erreichte Note in das Protokoll eintragen. Ferner wird, in dem Moment, in dem für eine bestimmte Identität eine Note eingetragen oder geändert wird, eine Benachrichtigungsemail an die betreffende Person versendet.

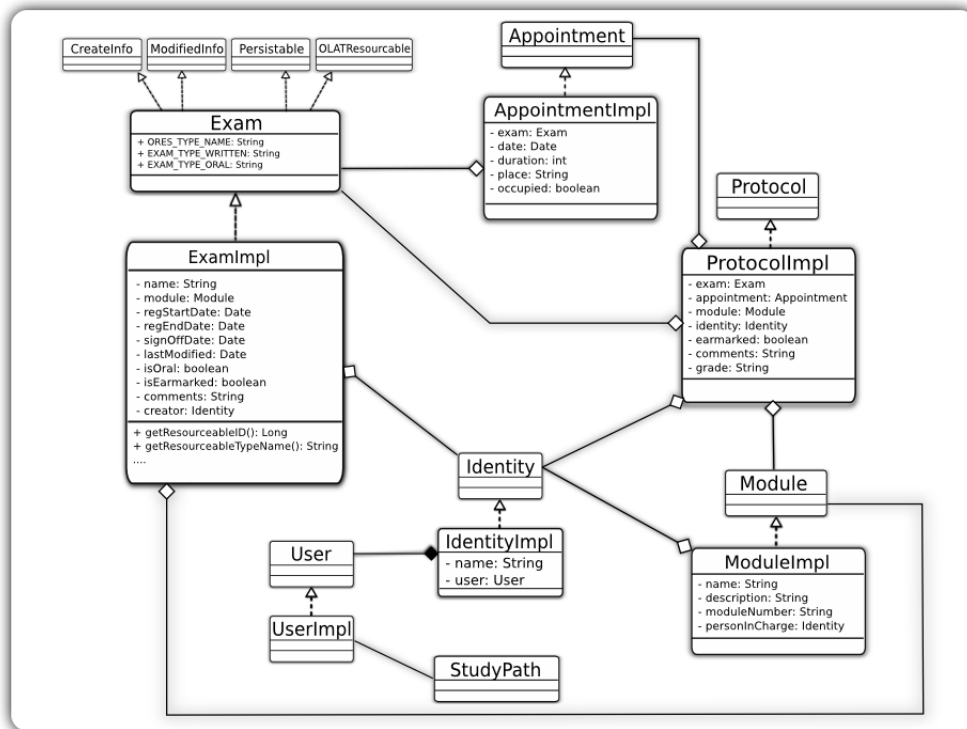


Abbildung 2: Klassendiagramm Prüfung

Zustand

Nachdem nun die statische Sicht auf eine Prüfung und die von ihr verwendeten Komponenten betrachtet wurde, wird in diesem Abschnitt die dynamische Sicht, insbesondere der Lebenszyklus einer Prüfungsinstanz, betrachtet. Als Ausgangspunkt dazu dient das in Abbildung 3 dargestellte Zustandsdiagramm, welches nun näher beschrieben wird.

Der Lebenszyklus einer Prüfung beginnt mit der Erstellung einer Prüfungsinstanz durch einen dafür autorisierten Benutzer. In der Regel sollten das die in einer Abteilung für den Prüfungsbetrieb zuständigen Angestellten (P-V) sein. Dieser erzeugt die Instanz, die zunächst nur die Art der Prüfung, die Modulzugehörigkeit sowie der Name selbst zugeordnet werden. Im Anschluss können nun auch die die Prüfung genauer beschreibenden Details wie die An- und Abmeldefrist, der Prüfungstermin sowie die verwendete Art der Einschreibung, etwa mit oder ohne Vormerken, spezifiziert werden. Ist dies geschehen und wurde der Termin vom Prüfungsamt akzeptiert, kann die Prüfung über den von OLAT vorgegebenen Publikationsprozess veröffentlicht und somit auch für alle Studenten sichtbar gemacht werden. Während des Einschreibezeitraums können sich nun Studenten mit validierter ESA in

die angebotenen Prüfungen einschreiben, sich vormerken und auch wieder austragen. Nach Ablauf des Einschreibezeitraums sind die Prüfungsanmeldungen verbindlich. Nun kann nur noch der Prüfungsverantwortliche oder das Prüfungsamt (am Portal registrierte) Kandidaten manuell ein- oder austragen. Solche manuell durchgeführten Ein- beziehungsweise Ausschreibungen werden in der elektronischen Studentenakte vermerkt. Zusätzlich kann die Prüfung bis zum Tag des tatsächlichen Stattfindens so editiert werden, dass An- und Abmeldezeiträume sowie alle anderen Attribute beliebig angepasst werden können. Nachdem die Klausur stattgefunden hat und die Korrektur der Prüfung abgeschlossen ist, können die Prüfungsergebnisse vom P-V in die seit der Anmeldung vorhandenen Tabellen eingetragen werden. Diese Eintragungen werden automatisch im Prüfungsprotokoll vermerkt. Der Prüfling wird per Email von der Bewertung in Kenntnis gesetzt und kann die eigenen Ergebnisse einsehen. Nun folgt die übliche Einsichtnahme der Studenten in ihre Prüfungen. Sollten etwaige Fehler bei der Benotung passiert sein, können die Noten entsprechend verändert und ebenfalls Kommentare im Prüfungsprotokoll vermerkt werden. Diese werden ebenfalls in der elektronischen Studentenakte eingetragen. Ist die Phase von möglichen Einsprüchen beziehungsweise Notenänderungen beendet, werden die Prüfungsergebnisse an das Prüfungsamt gesendet. Die Prüfung ist damit beendet und sollte nach einer Frist von einem Jahr gelöscht beziehungsweise extern archiviert werden.

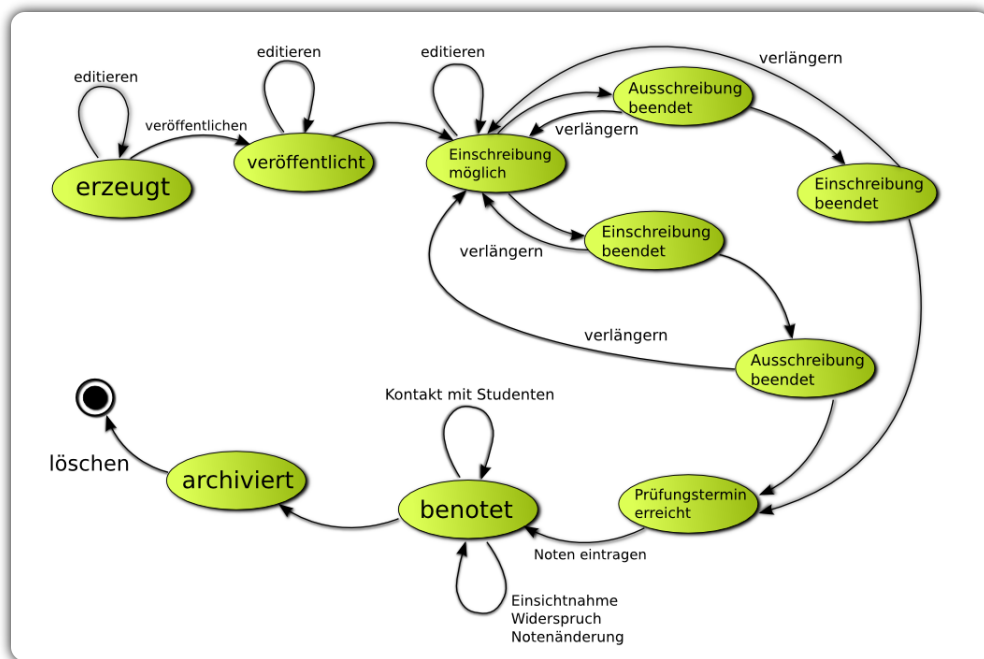


Abbildung 3: Zustandsdiagramm einer Prüfung

5.3 Die elektronische Studentenakte

Nachdem das Prüfungsverwaltungssystem XMAN im Wintersemester 2008/2009 zum ersten Mal erfolgreich zur administrativen Unterstützung einiger Prüfungen am Lehrstuhl für betriebliche Informationssysteme genutzt wurde, stellte sich heraus, dass zwar einzelne Prüfungen selbst gut durchzuführen waren, aber der zu der Zeit vorhandene Funktions-

umfang keinesfalls über diesen speziellen Arbeitsablauf hinaus einsetzbar war. So konnte beispielsweise nicht in einer einfachen und klar definierten Vorgehensweise nachvollzogen werden, welche Prüfungsleistungen ein einzelner Student bereits erbracht hat. Diese Informationen waren zu dieser Zeit über die verschiedenen Prüfungsinstanzen verteilt und an keiner Stelle aggregiert. Ein weiterer, noch gravierenderer Nachteil war, dass praktisch keinerlei Überprüfung der Identitäten durchgeführt wurde. So konnte sich zum Beispiel jede (fremde) Person am System registrieren und sich anschließend in anstehende Prüfungen einschreiben. Um diese Unzulänglichkeiten zu beheben und das Gesamtsystem für eine größere Anwenderschicht attraktiver zu gestalten, wurden iterativ Erweiterungen und Änderungen an XMAN vorgenommen. Das Ergebnis dieser Iterationen ist die elektronische Studentenakte, im Folgenden auch ESA genannt, und die Integration dieser in das bereits bestehende System. In den folgenden Abschnitten wird diese ESA vorgestellt und versucht, einen Überblick über das Zusammenspiel dieser Akte mit den anderen Komponenten des PVS zu vermitteln.

Beantragung und Validierung

Um eine ESA zu beantragen, ist es auf der Seite des Studenten nötig, sich am Prüfungsportal zu registrieren. Diese Registration geschieht über den von OLAT vorgegebenen Registrationsprozess, bei dem der am System eindeutige Loginname und ein den OLAT Vorgaben entsprechendes Passwort angegeben werden muss. Möchte ein Prüfling nun an einer Prüfung teilnehmen, das heißt sich zu einem bestimmten Termin einschreiben, so muss er seine persönliche Studentenakte beantragen und validieren lassen. Dazu müssen die Felder *Vorname*, *Nachname*, *Studserv-Emailadresse*, *Matrikelnummer* sowie *Studiengang* wahrheitsgemäß ausgefüllt werden. Anschließend wird vom System eine temporäre, also nicht validierte Studentenakte erzeugt und diese in der Liste der zu validierenden Akten angezeigt. Dieser Vorgang wird in Abbildung 4 in der Form eines Sequenzdiagramms dargestellt. Von nun an können dafür autorisierte Benutzer, also Mitarbeiter des Prüfungsamtes beziehungsweise Nutzer mit der Rolle des Prüfungsamtmitarbeiters, die von den Studenten eingegebenen Werte mit den fakultätsinternen oder eventuell auch fakultätsübergreifenden Listen vergleichen und gegebenenfalls Änderungen fordern oder sogar selbst vornehmen. Hierfür kann eine Email an den Studenten geschickt werden, welche in der Akte festgehalten wird. Außerdem haben Nutzer mit der Rolle des Benutzerverwalters zusätzlich die Möglichkeit, Attribute des Studenten über die von OLAT vorgegebenen Änderungsfunktionen zu aktualisieren. Sind alle Informationen des Studenten korrekt, kann die Akte validiert werden, wobei ein entsprechender Eintrag in der Akte abgelegt wird. Von nun an darf sich der Student zu beliebigen Prüfungen einschreiben.

Es ist darauf geachtet worden, dass Studenten unter dem Menüpunkt *Home > Einstellungen* keine persönlichen Attribute verändern können, welche die Korrektheit der Validierung der Studentenakte kompromittieren könnten. Diese Einstellungen werden in der von OLAT dafür vorgesehenen Datei *user_config.xml* vorgenommen. Sollten dennoch Änderungen wie Studiengangwechsel oder heiratsbedingte Änderung des Nachnamens nötig sein, geschieht dies ausschließlich über die Funktion *ESA > Änderung beantragen*. Hierbei werden die Informationen der Akte aktualisiert und diese zur erneuten Validierung dem Prüfungsamt vorgelegt.

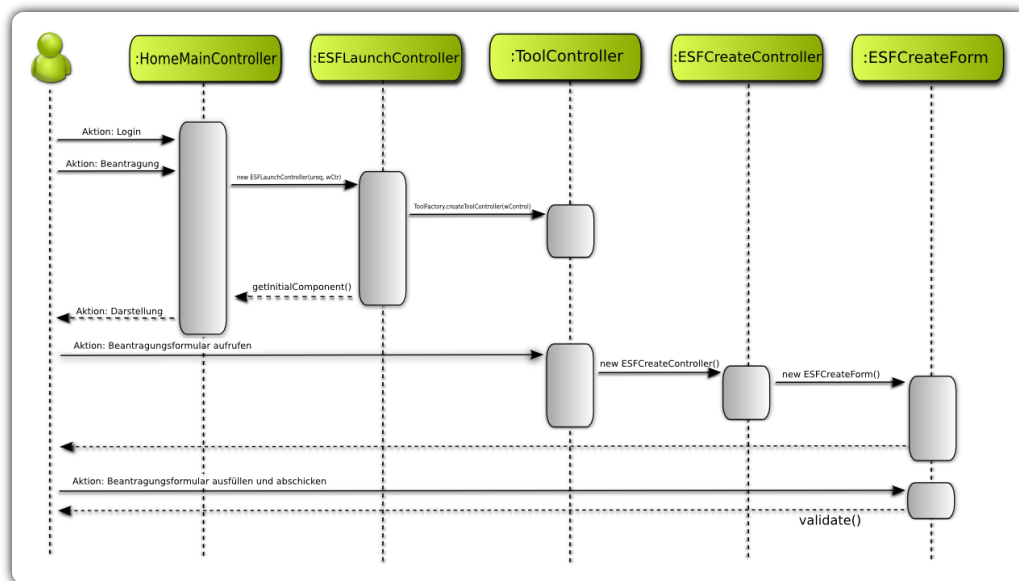


Abbildung 4: Sequenzdiagramm – Beantragung einer elektronischen Studentenakte

Aufbau

Um der realen Welt, also den in Papierform vorliegenden Studententaken des Prüfungsamts, möglichst genau zu entsprechen, wurde die elektronische Version, wie auch in den Anforderungen vorgegeben, in 4 Teile gegliedert. Dieser in Abbildung 5 veranschaulichte Aufbau wird nun im Folgenden beschrieben.

Persönliche Informationen Diese Art der Information werden, anders als alle restlichen Informationen, nicht von der Prüfungsverwaltungssoftware administriert. Es wird hier lediglich die Referenz auf die im OLAT-System registrierte Identität, also die des Studenten gespeichert. Hierüber ist es möglich, alle persönlichen Informationen über die von OLAT vorgegebenen Möglichkeiten zu speichern beziehungsweise zu aktualisieren. Außerdem wird dadurch vermieden, Daten redundant zu persistieren und somit auch die Konsistenz der Erweiterung gewahrt.

Prüfungsinformationen Wie bereits erwähnt, wurden Prüfungsergebnisse einzelner Studenten in den vorangegangenen Versionen dieser Prüfungsverwaltungssoftware nicht aggregiert. Dieser Nachteil wird nun an dieser Stelle ausgebessert und bietet somit einen Überblick über die bisherigen Prüfungsleistungen eines bestimmten Studenten. Es werden hier aus der Menge aller Protokolle diejenigen ausgewählt und angezeigt, die einer konkreten Identität zugeordnet sind. Zusätzlich wird ein Link zu den absolvierten Prüfungen und deren Kommentaren sowie Noten zur Verfügung gestellt.

Kommentare Kommentare zu einer ESA unterscheiden sich von den Kommentaren zu einer Prüfung. Hier geht es nicht um präzise Einträge zu einzelnen Klausuren, sondern vielmehr um Informationen, die das gesamte Studium eines Studenten betreffen, beispielsweise zu welchen Prüfungen er sich (nicht) eingeschrieben oder ob er den Studiengang gewech-

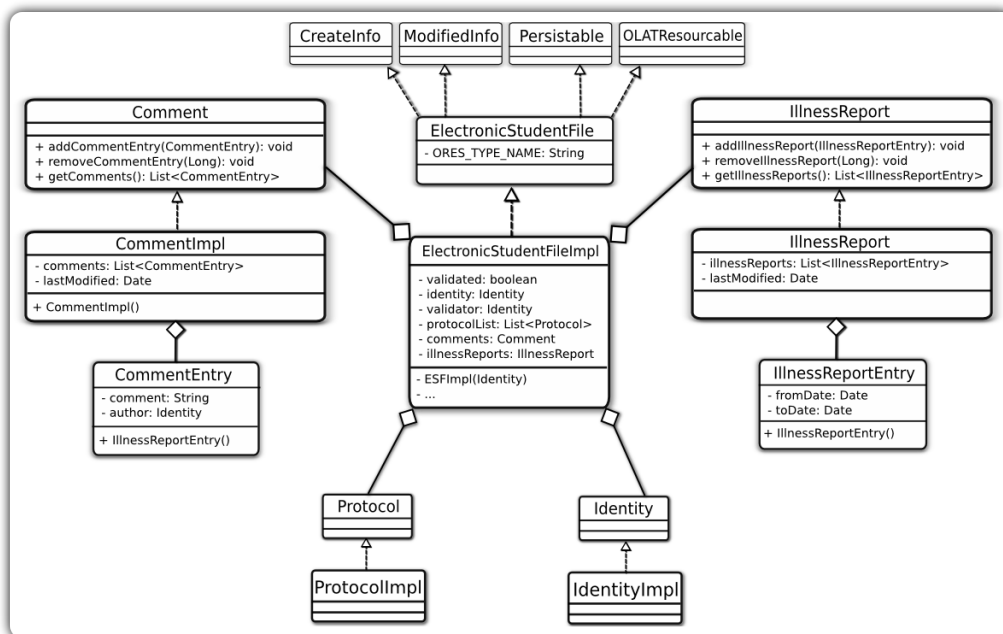


Abbildung 5: Klassendiagramm – Elektronischen Studentenakte

selt hat. Zusätzlich kann man hier noch zwischen automatisch generierten und manuell vom Prüfungsamt hinzugefügten Kommentaren unterscheiden, welche aber auf dieselbe Weise dargestellt sind.

Krankenscheine Mit Hilfe dieses Bausteins wird die elektronische Studentenakte komplettiert und Mitarbeitern des Prüfungsamtes die Möglichkeit gegeben, die auf Grund von Krankheit nicht wahrgenommenen Prüfungstermine zu verwalten. Bis jetzt wurde diese Funktion allerdings noch nicht praktisch eingesetzt, weshalb eine Beurteilung der Angemessenheit der Umsetzung dieser Komponente nicht möglich ist.

Administration der Studentenakten

Das Verwalten von elektronischen Studentenakten ist in XMAN Benutzern mit der Rolle des Prüfungsamtmitarbeiters und des Administrators vorbehalten. Somit kann sichergestellt werden, dass vertrauliche Informationen nur von vertrauenswürdigen Nutzern eingesehen werden können. Um die manuelle Valisierung der ESA möglichst intuitiv und schnell durchführen zu können, wurden Tabellen mit *MultiSelectionActions* für die Anzeige der Akten verwendet. Hierdurch können Aktionen wie *Validieren* für eine Vielzahl von Studenten gleichzeitig durchgeführt werden. Außerdem bietet die Funktion *Suchen*, die den *UserSearchController* von OLAT verwendet, eine einfache Möglichkeit, einzelne Akten auch bei einer Vielzahl am System registrierter Studenten zu finden. Andere für die Administration wichtige Funktionen wie das Löschen einer ESA oder die amtliche Email funktionieren ebenfalls für mehrere Personen gleichzeitig.

6 Implementierungsdetails

Der Fokus der bisherigen Abschnitte lag zu großen Teilen auf der Architektur der Prüfungsverwaltungssoftware. Um einen groben Überblick über die Verbindung und Abhängigkeiten zwischen OLAT und XMAN zu geben, wird in den folgenden Abschnitten ausschnittsweise die Verknüpfung der beiden Komponenten und die dazu verwendeten Techniken erläutert. Ferner ist zu bemerken, dass hier nicht die OLAT-Basistechnologien selbst erläutert werden, sondern nur deren Anwendung in der Erweiterung.

Manager Zu jedem Datenobjekt, wie der elektronische Studentenakte oder der Prüfung, existiert ein spezieller Manager, der auf diesen Objekten arbeitet. Diese Managerklassen folgen somit dem in [4] beschriebenen Prinzip, die Daten von den eigentlichen Datenzugriffsmechanismen zu separieren, wodurch die Datenobjekte leicht ausgetauscht werden können. Konkreter sind diese Manager für das Erzeugen und Löschen dieser Objekte zuständig, ohne dabei aber ihren internen Zustand zu kennen oder zu verändern. Außerdem implementieren diese Klassen auch die Schnittstelle zur relationalen Datenbank über Hibernate. So können sie etwa Objekte in der Datenbank aktualisieren oder komplexere Anfragen, wie zum Beispiel die Anzahl aller nicht validierten Studentenakten zu einem Studiengang, an dieselbe stellen. Zusätzlich sind alle Manager Singletons, das heißt, sie besitzen einen privaten Konstruktor und eine öffentliche sowie statische *getInstance()*-Methode. Somit ist gewährleistet, dass nur ein Managerobjekt zur gleichen Zeit mit einem Datenobjekt auf der Datenbank arbeitet.

Controller Wie durch das MVC Prinzip vorgegeben, existieren in den Paketen *admin*, *catalog*, *esf* und *exam* Ordner, die Controller enthalten. Um eine möglichst nah an OLAT angelegte Implementierung zu bieten, existieren für die beiden Datenobjekte Prüfung und elektronische Studentenakte jeweils Controller zum Erzeugen, Ändern und Starten dieser Ressourcen. Im Paket *admin* und *catalog* dient ein Controller nicht wie eben beschrieben zum Organisieren von Datenobjekten, sondern der Durchführung komplexer Arbeitsabläufe. So dient beispielsweise der Controller *ExamAdminESFController* zur Validierung von Studentenakten, zum Versenden von Emails, dem Suchen von Studentenakten und so weiter, also allen administrativen Aufgaben des Prüfungsamts. Ein einzelner Controller enthält im Wesentlichen einen Konstruktor, über den er beispielsweise seinen *ToolController* initialisiert, ein Velocity-Template lädt und eventuell benötigte Tabellen beziehungsweise Formulare erzeugt, und zwei *event*-Methoden. Die *Event*-Methoden dienen dazu, Ereignisse von anderen überwachten Objekten abzufangen und entsprechend zu behandeln. Die beiden *event*-Methoden unterscheiden sich lediglich in der Art der überwachten Objekt, die einerseits selbst wieder Controller sind und andererseits Tabellen oder Formulare. Das folgende Codebeispiel verdeutlicht, wie diese *event*-Methoden grundsätzlich arbeiten:

Anzeige OLAT verwendet für die Realisierung der MVC Architektur und der daraus resultierenden Trennung von Applikationslogik und Layout das unter [3] beschriebene GUI Framework. Um dieses Prinzip auch in den Erweiterungen aufrecht zu erhalten, werden alle GUI-Arbeitsabläufe, wie auch im vorherigen Abschnitt beschrieben, ebenfalls in wiederverwendbaren Controllern gekapselt, wodurch das Layout nur noch durch CSS-Modifikationen und HTML-Fragmente beeinflusst werden kann, wie auch Abbildung 7 zeigt. Es ist somit möglich, graphische Inhalte – in unserem Fall Tabellen und Formulare – in Controllern


```

1 // ein überwachter Controller hat ein Ereignis ausgelöst
2 public void event(UserRequest ureq, Controller ctr, Event event) {
3
4     // der das Ereignis auslösende Controller war der eigene ToolController
5     if ( ctr == this.toolCtr ) {
6
7         // das auslösende Ereignis ist: jemand möchte eine Studentenakte anlegen
8         if ( event.getCommand().equals("action.add.esf") ) {
9
10            ...
11        }

```

Abbildung 6: Funktionsweise der *Event*-Methoden

```

1 <!-- Tabelle mit Prüfungsprotokollen -->
2 <fieldset class="o_whiteBg"><legend><b>$r.translate("esf-edit_html.myExams")</b></legend>
3
4     $r.render("protocolTable")
5 </fieldset>

```

Abbildung 7: Auszug aus einem Velocity Template

zu generieren beziehungsweise deren Zustand zu berechnen und anschließend der *View*-Komponente zu übergeben, welche diese dann über den zugehörigen *Renderer* in eine HTML-Repräsentation überführt.

Literatur

- [1] D. Gerber: Operative Prozessunterstützung durch Integration einer Prüfungsverwaltung in das LMS OLAT. Bachelorarbeit, Univ. Leipzig, 2009.
<http://www.informatik.uni-leipzig.de/~graebe/Texte/Gerber-09.pdf>
- [2] OLAT – Online Learning And Training, ein web-basiertes Open Source Learning Management System. Projekt der Abteilung *Multimedia and E-Learning Services* an der Universität Zürich. <http://www.olat.org>
- [3] M. Stock, F. Jost, *OLAT Web GUI Framework*, JLS goodsolutions GmbH, 2006.
- [4] Sun Microsystems, Inc., *Core J2EE Patterns – Data Access Object*, 2002.