

## **Entwicklerdokumentation**

# **XML-Import/Export von Forum- und Wiki-Strukturen im OLAT**

### **Inhaltsverzeichnis**

1. Allgemeines.....	2
2. Produktübersicht.....	2
3. Grundsätzliche Struktur- und Entwurfsprinzipien für das Gesamtsystem.....	3
4. Grundsätzliche Struktur und Entwurfsprinzipien für einzelne Pakete.....	5
4.1 Funktionsweise der Controller.....	5

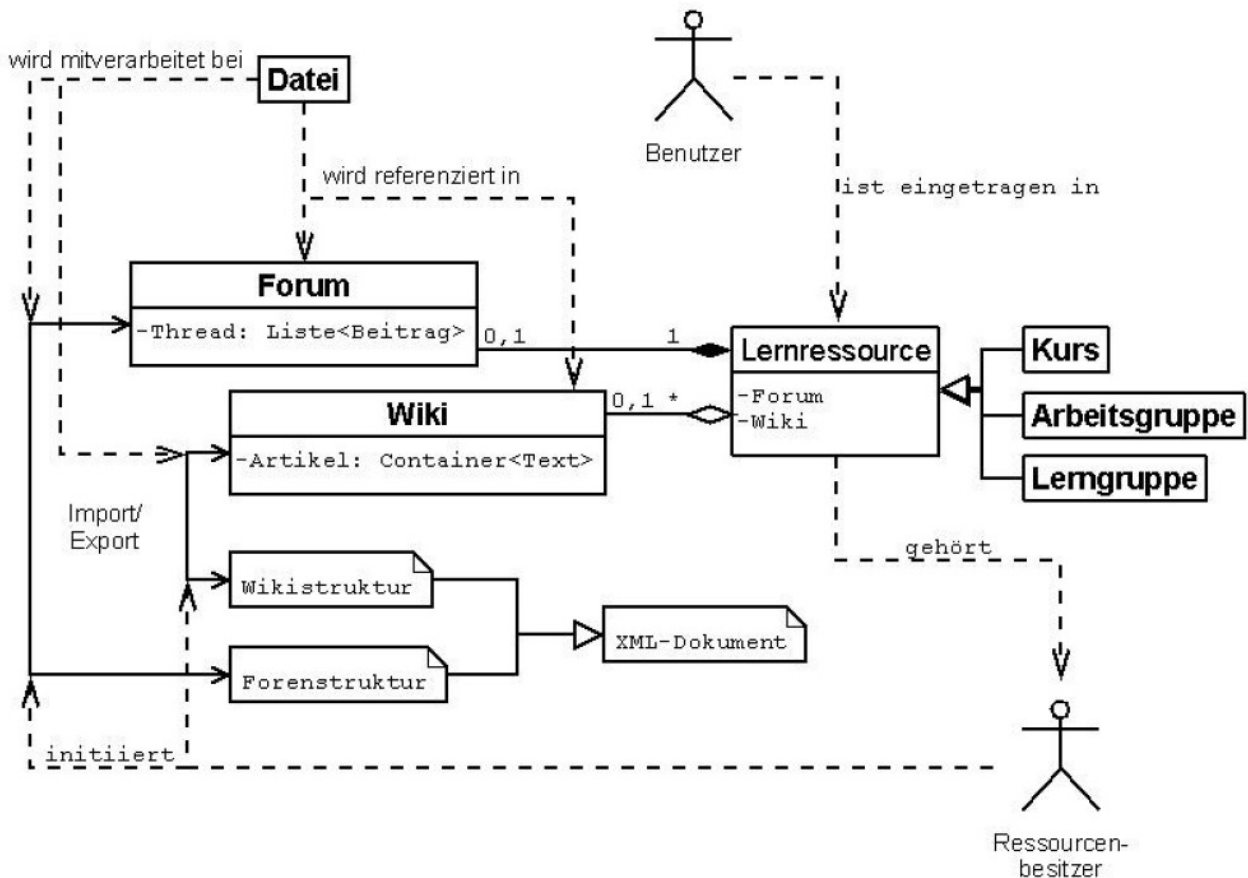
# 1. Allgemeines

OLAT stellt als E-Learning-Plattform viele Funktionen, wie Kollaborationen in Lerngruppen und Kursen, bereit. Diese können leicht verwaltet und durch verschiedene Komponenten erweitert werden, z.B. durch Wiki, Forum, Übungen und Tests.

Das vorliegende Projekt hat zum Ziel, das bestehende OLAT um die neuen Funktionen des Im- und Exportes von Foren- und Wiki-Inhalten in XML-Dateien zu erweitern. Im Wesentlichen wird dies durch das Hinzufügen von jeweils zwei Schaltflächen zu den jeweiligen Ansichten erreicht, die die Funktionen einfach nutzbar machen. Diese Schaltflächen sind für die Besitzer der Lernressource (des Kurses, der Lern-, Arbeitsgruppe zu dem/der das Forum oder Wiki gehört) sowie jeden Administrator sicht- und nutzbar.

# 2. Produktübersicht

Das Produkt ist dazu gedacht, den Im- und Export von Foren und Wikis aus OLAT-Lernressourcen zu realisieren, sodass sowohl die Portierung solcher Strukturen auf andere OLAT-Instanzen als auch eine bequeme Möglichkeit zur Sicherungskopie bereitstehen. Dazu werden diese vollautomatisch und mitsamt eingebundener Multimedia-Dateien in spezielle XML-Dateien überführt und später aus diesen ebenso pflegeleicht wiederhergestellt.



Besitzern einer Lernressource zeigt sich das Produkt in Form zweier neuer, unaufdringlicher Schaltflächen, die immer dann eingesehen (und damit betätigt) werden können, wenn der Nutzer auch die betreffende Struktur (das Forum oder Wiki) vor Augen hat. Die selbsterklärend mit „Exportieren nach XML“ und „Importieren aus XML“ beschrifteten Schaltflächen führen den Nutzer daraufhin zu einem erscheinenden Dialog. Im Falle eines Exportes ist die einzige nötige Eingabe des Nutzers ein simpler Druck auf die Schaltfläche „Fertigstellen“, um ein ganzes Forum oder Wiki in seinen persönlichen Ordner zu exportieren, beim Import muss er zuvor noch die Quelldatei für denselben über eine handliche Liste auswählen. Eventuell weitere zur Verfügung gestellte und erscheinende Optionen bleiben dabei völlig, ihrem Namen entsprechend, optional. Nur wenn ein Import von lokalen Rechnern durchgeführt werden soll, in dem Multimedia-Ressourcen vorkommen, ist eine aus internalen Gründen etwas aufwändigere Bedienung notwendig, da dann Forum/Wiki und Multimedia-Daten in getrennten Dateien gelagert werden müssen und entsprechend beide zu importieren sind. Entsprechende Anleitung wird dem Benutzer aber selbstverständlich im betroffenen Dialog geboten.

Ist dem Wunsch des Nutzers erst genüge getan, informiert ihn ein Popup darüber, dass dies geschehen ist, andernfalls, dass es nicht möglich war.

Selbstverständlich ist es dem Nutzer jederzeit außer bei der eigentlichen Durchführung der von ihm gewünschten Aktion möglich, den Prozess konsequenzlos abubrechen, worauf er zu genau jenem Fenster gelangt, das er betrachtet hatte, als er die Schaltfläche des Produktes betätigte.

### **3. Grundsätzliche Struktur- und Entwurfsprinzipien für das Gesamtsystem**

Das Produkt ist auf höchster Ebene nach der MVC-Architektur gegliedert. Die Pakete „partly“ und „converter“ kapseln hierbei die Modellfunktionalitäten, während das Paket „wizards“ Viewfunktionalitäten enthält. Controller ist „controller“.

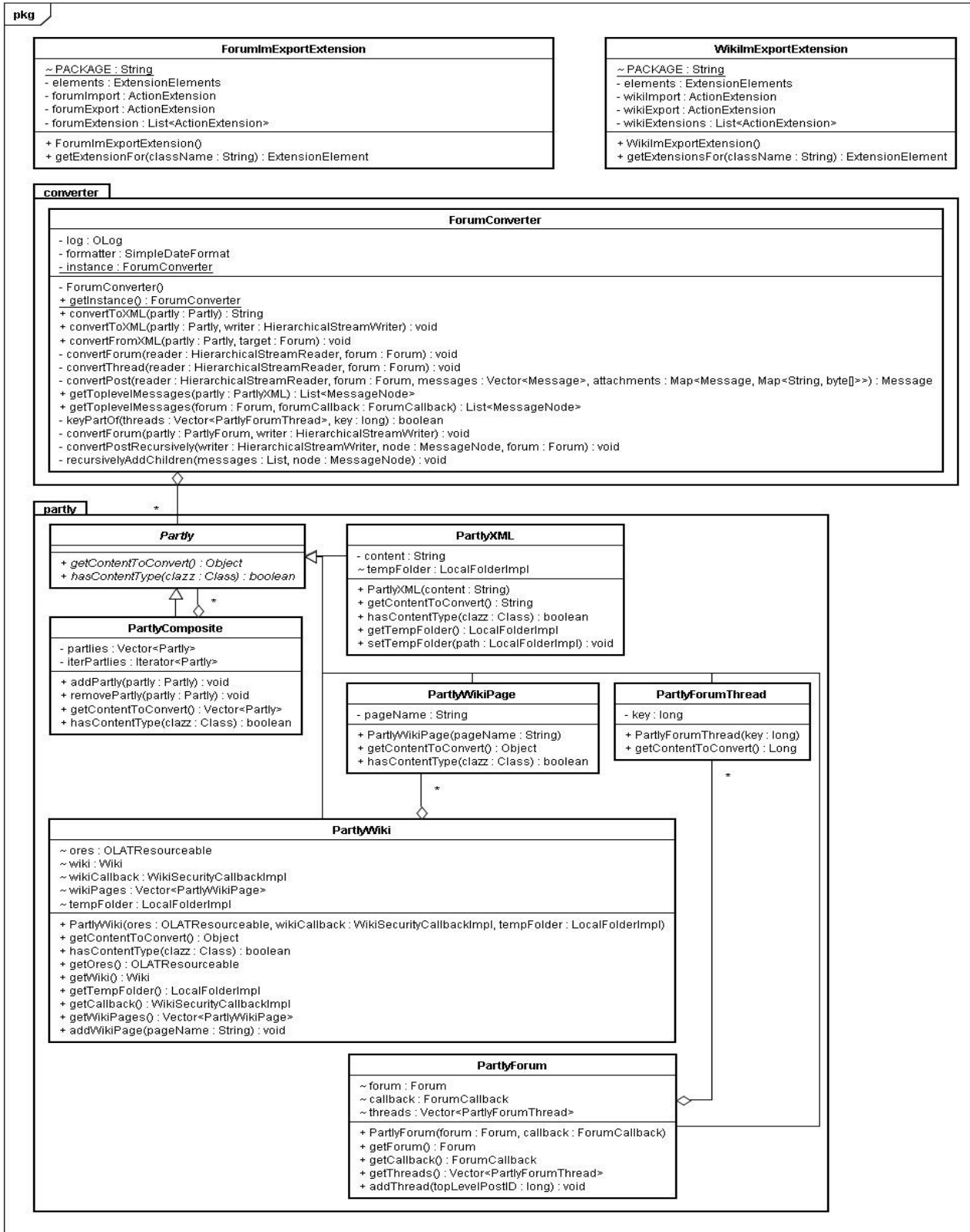
Auch ist durch Einschränkung der Sichtbarkeit all jener Klassen, die teil eines Paketes sind und nur innerhalb desselben Arbeit zu verrichten haben, das „BlackBox“-Prinzip dergestalt realisiert, dass nur die nötigsten Bestandteile jedes Paketes der Außenwelt zugänglich sind.

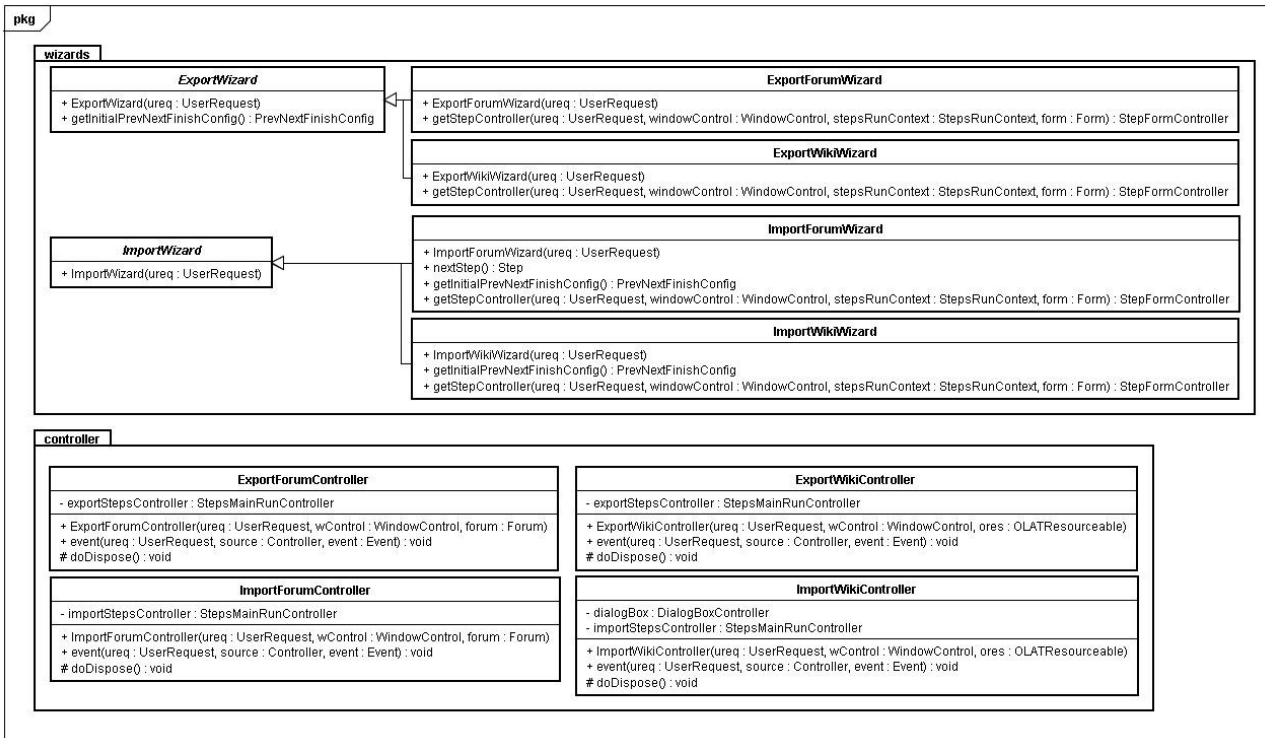
Da in OLAT selbst die MVC-Architektur<sup>1</sup> mit dem Ziel größerer Zweckmäßigkeit abgewandelt ist, erscheint es nur natürlich, dass auch das Produkt, als OLAT-Erweiterung, dieser Abwandlung folgt. So ist das Modell beispielsweise nicht nur mit der simplen Verarbeitung der ihm zugewiesenen Daten betraut sondern auch damit, sie durch Übergabe an zutreffende OLAT-Manager an die gewünschte Stelle zu befördern.

---

<sup>1</sup> siehe Entwurfsbeschreibung des Fremdprojektes

*Übersicht über die verwendeten Klassen:*





## 4. Grundsätzliche Struktur und Entwurfsprinzipien für einzelne Pakete

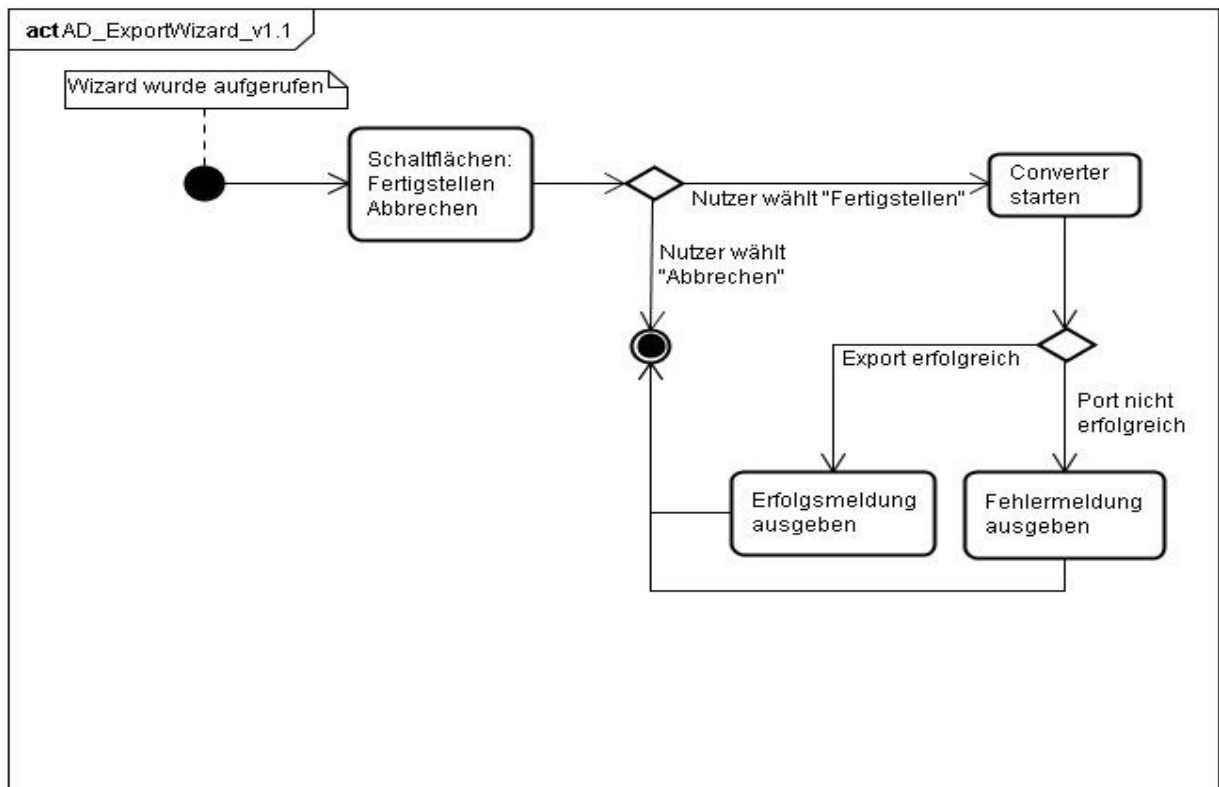
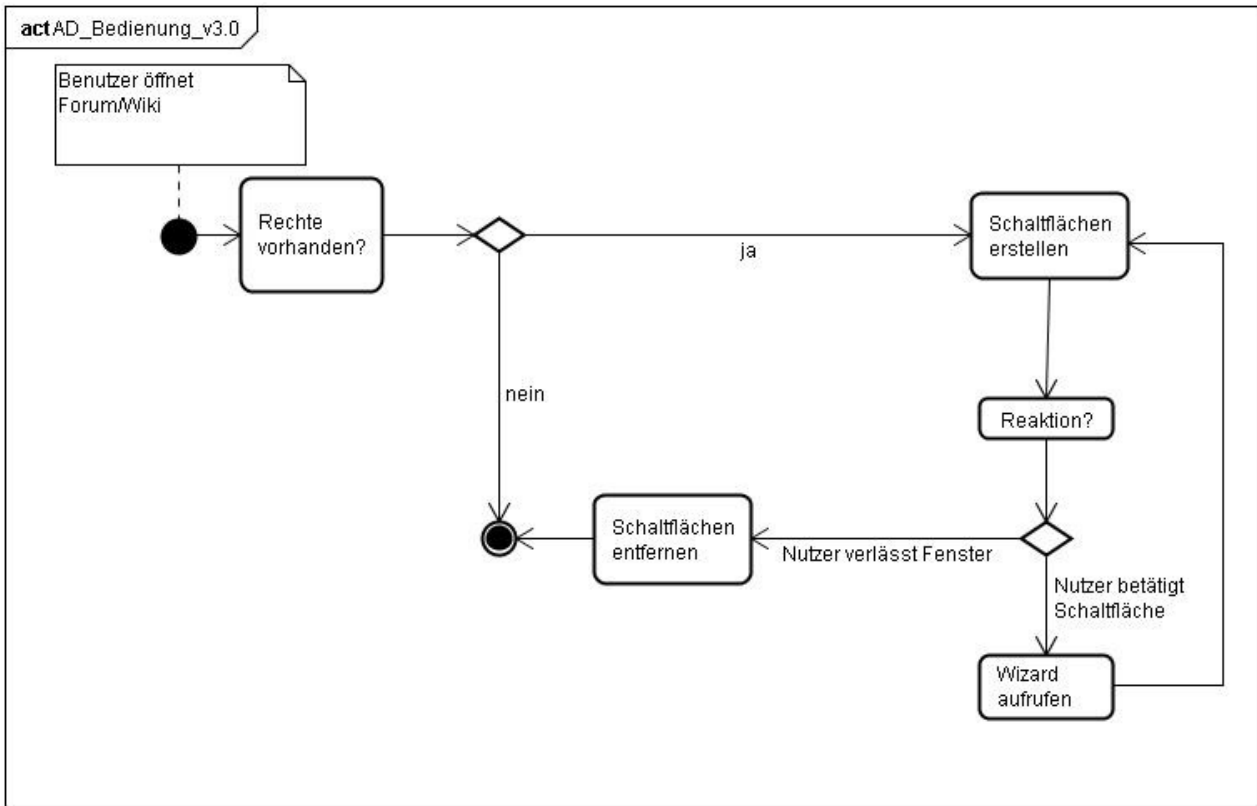
### 4.1 Funktionsweise der Controller

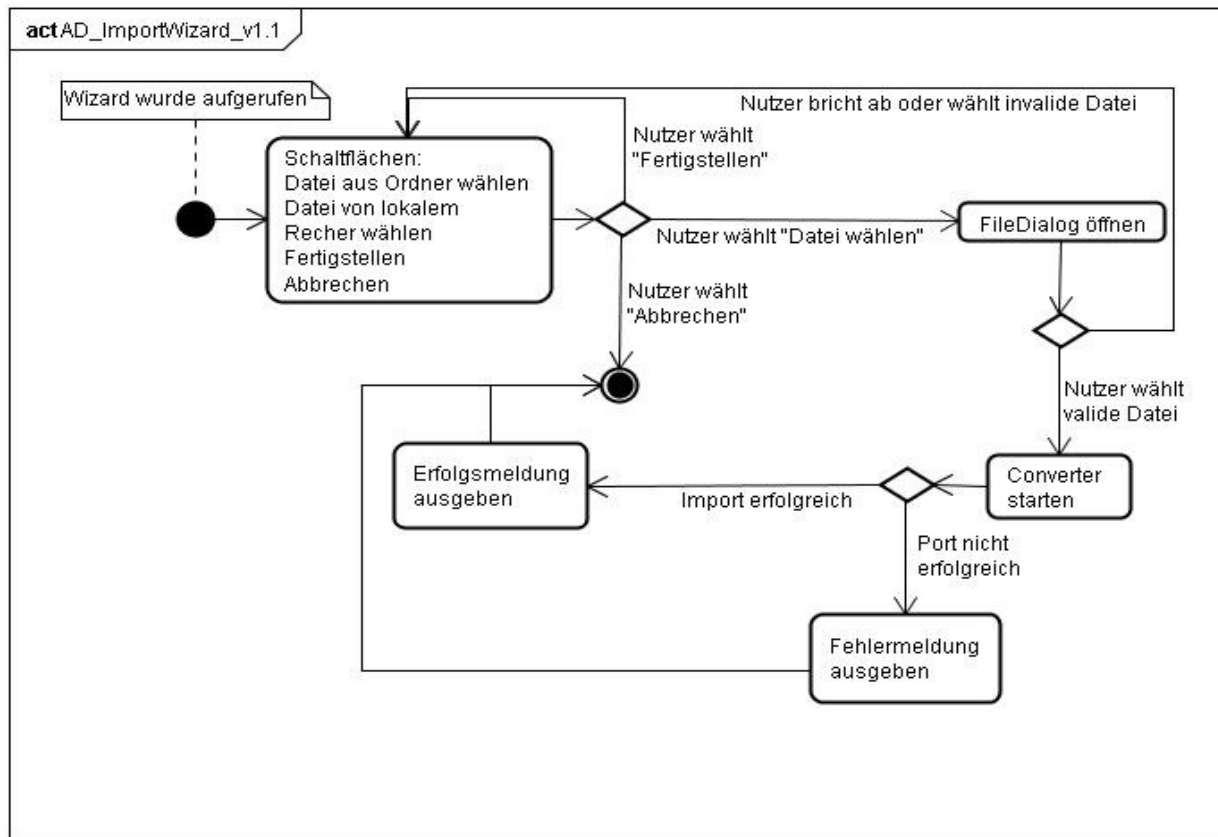
Die Controller sind für die Interaktion mit dem Benutzer verantwortlich. Falls ein Benutzer entsprechende Rechte besitzt, wird eine der Controllerklassen zusammen mit zugehörigen Schaltflächen vom *ForumController* beim Aufrufen eines Forums sowie vom *WikiMainController* beim Aufrufen eines Wikis erstellt. Für erwähnte Schaltflächen wird auf das *Component*-System des OLAT-Framework zurückgegriffen. Neben deren Funktionalität nimmt der erzeugte Controller auch eventuell später vom Benutzer hervorgerufene Ereignisse entgegen, die mit dem Produkt in Verbindung stehen.

Export und Import von Strukturen wie Forum oder Wiki sind komplex, sodass ein Teil der Funktionalität ausgelagert werden soll. Über eine Instanz seiner assoziierten Wizardklasse kann der Controller die Benutzerabfragen für den konkreten Vorgang ohne größeren Aufwand abwiegeln.

Der *Wizard* soll dann ein geeignetes Objekt vom Typ *Partly* anlegen und zurückliefern, mit dem eine Teilstruktur abgebildet wird, also ein Teil des Forums bzw. ein Teil des Wikis (falls ein Export durchgeführt wird) oder aber ein Teil einer XML-Struktur (falls importiert wird).

Folgende Aktivitätsdiagramme zeigt die Entscheidungen, die dabei getroffen werden:





Die *Partly*-Klassen sind nach dem Design-Pattern Kompositum aufgebaut, sodass hierarchische Strukturen (wie es bei XML, aber auch bei einem Forum oder Wiki der Fall ist) leicht abgebildet werden können. Es gibt konkrete *Partly*-Spezialisierungen, um Forum-Threads, Wiki-Artikel und XML-Tags zu speichern, aber auch allgemeine, von *Partly* abgeleitete Klassen, die es ermöglichen, mehrere *Partly*s zu einem einzigen zusammenzusetzen.

Die Aufgabe der *Converter*-Klassen ist das Abbilden von Objekten in XML-Strukturen bzw. das Auslesen von XML-Daten in konkrete Objekte. Um an die bereits geparste XML-Struktur zu gelangen, ist es zweckmäßig, auf bereits vom OLAT verwendete Bibliotheken wie xpp3 und xstream zurückzugreifen.

Für die Umwandlungsprozesse ist eine intensive Kommunikation mit der Daten-Schicht des OLAT notwendig, auf welche die *Converter*-Klassen bequem über die *Manager* zugreifen können.