Digitale Rätoromanische Chrestomathie

Werkzeuge und Verfahren für die kollaborative Volltexterschließung digitaler Sammlungen

Claes Neuefeind, Fabian Steeg

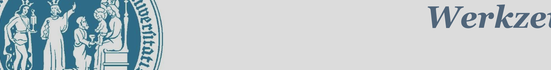
Universität zu Köln, Sprachliche Informationsverarbeitung

Überblick

Ziel des Projekts ist die Bereitstellung von Werkzeugen und Verfahren für die kollaborative Volltexterschließung digitaler Sammlungen am Beispiel der Rätoromanischen Chrestomathie. Dieses Poster beschreibt die Umsetzung der kollaborativen Korrekturumgebung, von der architektonischen Grundlage und technologischen Umsetzung, über den aktuellen Stand der Software zu der geplanten Weiterentwicklung im zweiten Projektjahr. Die Entwicklung erfolgt seit Beginn offen unter *http://github.com/spinfo/drc*

Architektur

Der Natur des Projekts wird eine dreischichtige Systemarchitektur gerecht: Gesamtziel des Projekt ist die Produktion bestimmter Daten (der korrigierten Texte). Diese Daten sind für alle Benutzer des Systems identisch, und können daher zentral gespeichert werden (Datenschicht). Verschiedene Nutzer sollen unabhängig voneinander auf diese Daten zugreifen, und diese verändern können, wobei die Integrität der Daten gewährleistet werden muss (Logikschicht). Der Zugriff erfolgt über eine graphische Benutzerschnittstelle (Präsentationsschicht).



Grundlegende Systemarchitektur

So ergibt sich eine dreischichtige Systemarchitektur mit Präsentations-, Logik- und Datenschicht. Dabei kommuniziert die Präsentationsschicht mit der Logikschicht und die Logikschicht mit der Datenschicht. Es gibt keine direkte Verbindung zwischen Präsentations- und Datenschicht. Auf diese Weise ist das System lose gekoppelt und erlaubt so Austausch und Wiederverwendung der Schichten, etwa für eine Aufbereitung der Daten in anderen Kontexten.

Technologie

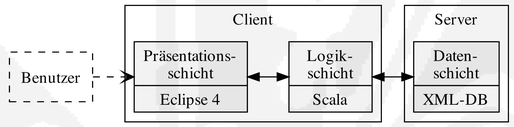
Zur Umsetzung der einzelnen Schichten bieten sich verschiedene Lösungen an, so kann die Präsentationsschicht als Web- oder als Desktop-Anwendung umgesetzt werden, die Datenschicht kann in Form von Dateien oder als Datenbank umgesetzt werden, und die Logikschicht kann innerhalb einer Client-Server-Architektur als Teil des Clients oder des Servers umgesetzt werden.

Editor

Aufgrund des modernen Programmierkonzepts, der hohen Modularität und Wiederverwertbarkeit durch OSGi, der nativen GUI-Technologie sowie der Integration von Webstandards (z.B. CSS zur Gestaltung der GUI), haben wir uns für Eclipse 4 (*http://eclipse.org/eclipse4/*) als Technologie für die Präsentationsschicht entschieden. Für eine kompakte und zugleich effiziente und kompatible Logikschicht setzen wir auf die JVM-Sprache Scala (http://www.scala-lang.org/). Aus Benutzersicht stellt sich die erste Beta-Version folgendermassen dar:

Screenshot der ersten Beta des Editors

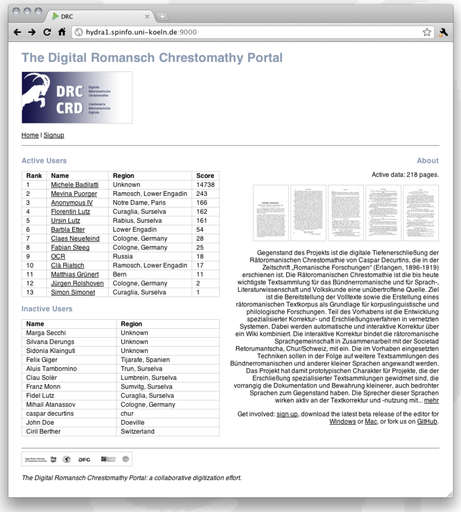
Für die Datenschicht haben wir in ersten Prototypen einfache XML-Dateien eingesetzt, die wir für die erste Beta-Version durch eine XML-Datenbank (*eXist, http://exist.sourceforge.net/*) ersetzt haben. Da *eXist* über eine eingebaute Serverfunktionalität verfügt war es zweckmässig, die Logikschicht als Teil des Clients umzusetzen, und so keine eigenen serverseitigen Komponenten implementieren zu müssen. Für die erste Beta-Version ergibt sich damit die folgende technologische Umsetzung der oben skizzierten Architektur:



Implementierung der Architektur in der ersten Beta

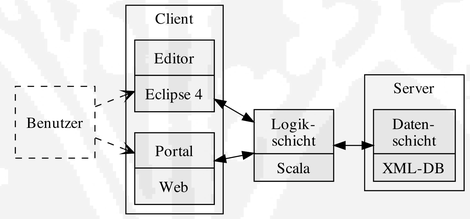
Portal

Für die zweite Beta soll eine Portalseite bereitgestellt werden, die als zentraler Anlaufpunkt für interessierte Nutzer fungieren soll und über die bestimmte Aspekte der bearbeiteten Daten dargestellt werden können (z.B. Anzahl der Nutzer, korrigierte Seiten, vergebene Schlagworte, etc.). Die Portal-Seite sieht zur Zeit folgendermassen aus:



Screenshot der Portalseite

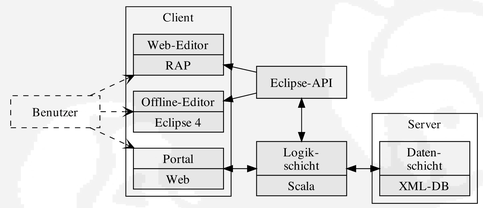
Die grundlegende Funktionalität der ersten Beta-Version wird in der zweiten Beta um zentrale Aspekte erweitert. Dies sind insbesondere die Nutzung der vorhandenen Göttinger Metadaten in der Benutzerschnittstelle (etwa zur Gliederung der Texta nach Bänden, etc.) und die Integration weiterer Bände der RC, sowie die Integration neuer Metadaten, die automatisiert aus dem Registerband gewonnen werden sollen. Daneben soll die Benutzerverwaltung ausgebaut werden (Benutzerprofile, Statistiken, etc.). Der Editor wurde zudem mit automatischen Aktualisierungen versehen, die es ermöglichen, neue Funktionalität und Fehlerbehebungen in der Software ohne großen Aufwand seitens der Nutzer bereitzustellen. Für die aktuelle zweite Beta-Version ergibt sich damit die folgende technologische Umsetzung der oben skizzierten Architektur:



Implementierung der Architektur in der zweiten Beta

Weiterentwicklung

Im weiteren Verlauf des zweiten Projektjahres sollen alternative Umsetzungen und ein Ausbau der GUI untersucht werden. Die aktuelle Umsetzung als Eclipse-basierte Desktop-Applikation, die als Client des Datenbankservers fungiert, ermöglicht sowohl eine Weiterentwicklung zu einer Offline-fähigen Desktop-Applikation, die ohne Netzzugang verwendet werden kann und bei Bedarf die Daten mit dem Server synchronisiert, als auch die automatische Generierung einer Web-Oberfläche aus der gleichen Code-Basis mithilfe des RAP-Frameworks (*http://eclipse.org/rap/*).



Implementierung der Architektur in einer zukünftigen Version