

Hochschule Bochum

Fachbereich Elektrotechnik und Informatik

# Arbeitsthese

**UML2 Web-Modelling-Tool**

**erstellt von:** Tim Keller <tim.keller@fh-bochum.de>

**erstellt am:** 26. September 2007

**Kooperation:** innoQ Deutschland GmbH

# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b>	<b>3</b>
<b>2</b>	<b>Erste Analyse</b>	<b>3</b>
2.1	Motivation . . . . .	3
2.2	Grundidee . . . . .	3
2.2.1	Generelles Vorgehen aus Benutzersicht . . . . .	3
2.3	Ziel der Diplomarbeit . . . . .	5
<b>3</b>	<b>Zielbestimmung</b>	<b>5</b>
3.1	Musskriterien . . . . .	5
3.2	Wunschkriterien . . . . .	6
3.3	Abgrenzungskriterien . . . . .	6
<b>4</b>	<b>Systemübersicht</b>	<b>7</b>
4.1	Client . . . . .	7
4.2	Server (eigentliche Applikation) . . . . .	8
4.2.1	Warum wird gerade Ruby on Rails verwendet? . . . . .	8
4.2.2	Was "beinhaltet" Rails? . . . . .	8
4.3	Datenbank . . . . .	8

# 1 Vorwort

Die Arbeitsthese beinhaltet die eigentliche Grundidee des Programms, eine kurze Darstellung der einzelnen Systemkomponenten und einen ersten UI-Entwurf.

## 2 Erste Analyse

### 2.1 Motivation

Meine Motivation ist, ein einfach zu bedienendes, überall erreichbares, plattformunabhängiges, von relativ neuen Technologien - Ruby on Rails - gebrauchmachendes UML-Tool zu schreiben, welches man in einem Webbrowser bedienen kann. Ein weiterer Ansporn ist, sich mit Programmierparadigmen wie dem MVC-Pattern (Model-View-Controller) und objektrelationalen Mapping auseinanderzusetzen.

### 2.2 Grundidee

Mit dem UML2 Web-Modelling-Tool - auch Produkt oder Programm genannt - soll ein Benutzer die Möglichkeit erhalten, ohne Installation zusätzlicher Plugins, außer aktiviertem JavaScript, über einen aktuellen Webbrowser (vorzugsweise Firefox<sup>1</sup> ab Version 2.0) Klassendiagramme nach der UML2<sup>2</sup> zu erstellen.

#### 2.2.1 Generelles Vorgehen aus Benutzersicht

Als erstes gibt der Benutzer die Adresse in seinen Webbrowser ein an welcher sich die Applikation befindet. Daraufhin gelangt er auf die Startseite des Programms. Eine fehlerfreie Kommunikation zwischen Client und Sever wird hierbei vorausgesetzt.

Nach einem erfolgreichen Login - jeder Benutzer besitzt ein Konto welches auch zur Authentifizierung dient - gelangt er zu seinem persönlichen Arbeitsbereich.

---

<sup>1</sup><http://www.mozilla-europe.org/de/products/firefox/>

<sup>2</sup><http://www.uml.org/>

Hier hat der Nutzer nun die Möglichkeit:

- neue Diagramme zu erstellen und zu speichern
- schon erstellte Diagramme zu laden und zu bearbeiten
- ältere Diagramme zu löschen

Ein erster UI-Entwurf:

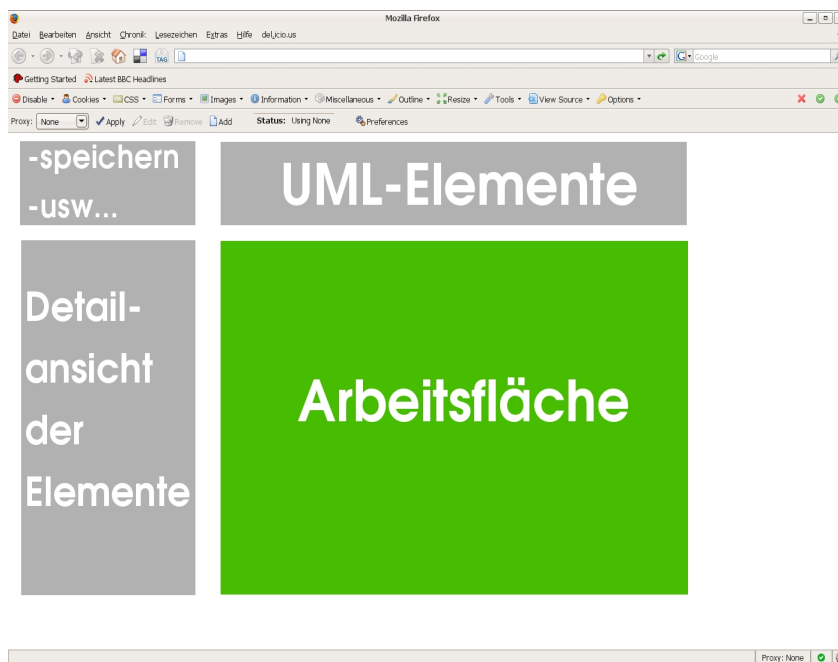


Abbildung 1: erster UI-Entwurf

Entscheidet sich der Benutzer ein neues Diagramm zu erstellen, wird die Arbeitsfläche freigegeben. Nun kann er mittels "Drag and Drop" Elemente aus der oberen Leiste in die Arbeitsfläche einbinden oder auch Elemente aus der Arbeitsfläche entfernen. Der Benutzer kann Elemente miteinander verbinden und beispielsweise in Klassen, Methoden und Attribute angeben. In der Detailansicht erhält er eine Übersicht der jeweiligen Klassen. Zum Schluss kann der Nutzer das Diagramm abspeichern und zu einem späteren Zeitpunkt wieder bearbeiten.

## 2.3 Ziel der Diplomarbeit

Im Mittelpunkt der Diplomarbeit steht der Umgang mit dem Web-Framework Ruby on Rails<sup>3</sup> und die Entwicklung des UML2 Web-Modelling-Tool. Ein weiterer, wichtiger Teil ist die Einarbeitung in die UML2-Spezifikation.

# 3 Zielbestimmung

## 3.1 Musskriterien

- der Benutzer kann Klassen - durch Rechtecke dargestellt - in die Arbeitsfläche einbinden
- die Klassen können verbunden werden
- der Benutzer kann Variablen und Methoden in eine Klasse eintragen
- der Benutzer kann Kommentare einfügen
- Diagramme können abgespeichert werden
- Diagramme erhalten eine eigene URL
- Diagramme können in das PNG oder JPG-Format umgewandelt werden
- XMI-Darstellung der Diagramme
- abgespeicherte Diagramme können geladen und bearbeitet werden
- das Programm läuft unter Firefox ab Version 2.0 auf einem Linux/Unix-System fehlerfrei

---

<sup>3</sup><http://www.rubyonrails.org/>

## 3.2 Wunschkriterien

- die Diagrammtypen sind UML2 konform
- Paketdiagramme werden unterstützt
- Kollaboration an Diagrammen ist möglich
- das Programm läuft unter anderen Browsern und Betriebssystemen fehlerfrei

## 3.3 Abgrenzungskriterien

- im Programm wird nur ein geringer Teil(Klassendiagramme evtl. Paketdiagramme) der UML2 abgebildet
- aus dem Diagramm wird kein Programm-Code erzeugt
- es wird kein eigenes Datenaustauschformat geschrieben, sondern bestehende Formate beispielsweise JSON<sup>4</sup> (JavaScript Object Notation) verwendet
- es wird auf bestehende JavaScript-Frameworks zurückgegriffen

---

<sup>4</sup><http://www.json.org/>

## 4 Systemübersicht

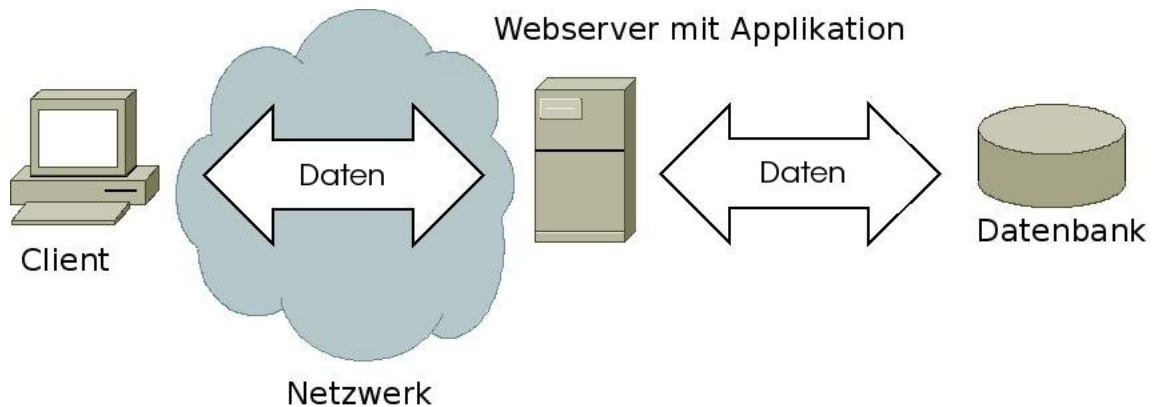


Abbildung 2: grober Systemaufbau

### 4.1 Client

Die Aufbereitung der Oberfläche wird mit Hilfe des JavaScript-Frameworks Prototype<sup>5</sup> bzw. script.aculo.us<sup>6</sup> realisiert. Der Benutzer muss lediglich JavaScript im Browser aktiviert haben, Plugins oder sonstige Programme sind zur Nutzung des Programms nicht erforderlich. Die Daten, also Koordinaten der Klassen, Verbindungen der einzelnen Klassen, Attribute, Methoden usw. werden mittels AJAX (Asynchronous JavaScript and XML)<sup>7</sup> im “Hintergrund” laufend an den Server gesendet. Sobald der User auf “speichern” klickt wird das Diagramm in der Datenbank abgespeichert oder Änderungen an einem vorhandenen Diagramm übernommen. Möchte der Benutzer Diagramme bearbeiten oder begutachten, werden die Daten vom Server geladen und im Browserfenster dargestellt. Zum Austausch der Daten zwischen Server und Client werden gängige Techniken wie das Datenaustauschformat JSON eingesetzt.

<sup>5</sup><http://www.prototypejs.org/>

<sup>6</sup><http://script.aculo.us/>

<sup>7</sup><http://developer.mozilla.org/en/docs/AJAX>

## 4.2 Server (eigentliche Applikation)

Die Applikation ist in der Programmiersprache Ruby geschrieben und macht Gebrauch von dem Web-Framework Ruby on Rails.

### 4.2.1 Warum wird gerade Ruby on Rails verwendet?

“**Web development that doesn’t hurt:** Ruby on Rails is an open-source web framework that’s optimized for programmer happiness and sustainable productivity. It lets you write beautiful code by favoring convention over configuration.”<sup>8</sup>.

### 4.2.2 Was “beinhaltet” Rails?

“Rails is a full-stack framework for developing database-backed web applications according to the Model-View-Control pattern. From the Ajax in the view, to the request and response in the controller, to the domain model wrapping the database, Rails gives you a pure-Ruby development environment. To go live, all you need to add is a database and a web server.”<sup>9</sup>.

## 4.3 Datenbank

Das relationale Datenbankverwaltungssystem MySQL<sup>10</sup> wird für die Datenerhaltung verwendet. Die Daten aus der Rails-Anwendung werden mithilfe von ActiveRecord - hier ist beispielsweise das objektrelationale Mapping oder auch der Zugriff auf spezielle Datenbanksysteme definiert - in der Datenbank abgespeichert.

---

<sup>8</sup>Zitat von <http://www.rubyonrails.org>

<sup>9</sup>Zitat von <http://www.rubyonrails.org>

<sup>10</sup><http://www.mysql.com>