



FAKULTÄT  
FÜR INFORMATIK  
Faculty of Informatics



# Online Medical Imaging Platform

Willinger Christin

Computer Vision Lab  
Institute of Computer Aided Automation  
Vienna University of Technology

July 2, 2014

Supervisor: Robert Sablatnig

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.1.1	Hardware und OS Abstraktion . . . . .	1
1.1.2	Interaktion mit einer Bildsuchmaschine . . . . .	1
1.1.3	Interface für Kreshmoi . . . . .	1
1.2	Möglichkeiten zur Umsetzung . . . . .	1
1.2.1	JavaApplet . . . . .	1
1.2.2	HTML5 . . . . .	1
1.3	Pflichtenheft . . . . .	1
<b>2</b>	<b>Related Work</b>	<b>2</b>
2.1	Funktionsumfang von Betrachtungstools . . . . .	2
2.1.1	Beispiel Osirix . . . . .	2
<b>3</b>	<b>Methodology</b>	<b>3</b>
3.1	Verwendete Technologien und Protokolle . . . . .	3
3.1.1	HTTP . . . . .	3
3.1.2	REST . . . . .	3
3.1.3	JSON . . . . .	3
3.1.4	AJAX . . . . .	3
3.1.5	Objectiv J . . . . .	3
3.1.6	Cappuccino . . . . .	4
3.2	Kommunikation mit KRESHMOI . . . . .	4
3.2.1	Query nach Bildern . . . . .	4
3.2.2	Laden der Bilder . . . . .	4
3.3	Architektur und Komponenten . . . . .	4
3.3.1	Domänen Model . . . . .	4
3.3.2	Architektur und Aufteilung in Komponenten . . . . .	4
3.3.3	2DView . . . . .	4
3.3.4	Kommunikations Module . . . . .	4
3.4	Usability . . . . .	4
3.4.1	Workflow bei der Befundung . . . . .	4
<b>4</b>	<b>Results</b>	<b>5</b>
4.1	Geschwindigkeit . . . . .	5
4.2	Usability . . . . .	5
<b>5</b>	<b>Conclusion</b>	<b>6</b>

## Abstract

# 1 Introduction

## 1.1 Motivation

### 1.1.1 Hardware und OS Abstraktion

### 1.1.2 Interaktion mit einer Bildsuchmaschine

### 1.1.3 Interface für Kreshmoi

## 1.2 Möglichkeiten zur Umsetzung

### 1.2.1 JavaApplet

### 1.2.2 HTML5

## 1.3 Pflichtenheft

## 2 Related Work

### 2.1 Funktionsumfang von Betrachtungstools

#### 2.1.1 Beispiel Osirix

## 3 Methodology

### 3.1 Verwendete Technologien und Protokolle

#### 3.1.1 HTTP

HTTP (Hyper Text Transfer Protokoll) ist ein Protokoll zur Übertragung von Daten über ein Netzwerk welches auf TCP aufsetzt. Der Datenaustausch findet zwischen zwei Kommunikationspartnern in der Form von Nachrichten statt, wobei der Client eine Anfrage an einen Server stellt und dieser die Anfrage bearbeitet und eine Antwort retourniert.

Eine Nachricht setzt sich aus einem Header und einem Body zusammen. Der Body enthält die Nutzdaten und der Header enthält Metadaten über die Nutzdaten. Vom Aufbau der Nachricht unterscheiden sich Anfrage und Antwort nur in der ersten Zeile:

- Anfrage: Enthält die HTTP-Methode, die URL welche auf die Resource am Server zeigt und die Protokollversion.
- Antwort: Enthält die Protokollversion und den Serverstatus. Der Serverstatus liefert eine Aussage ob der Request erfolgreich bearbeitet wurde bzw welche Art von Fehler bei der Bearbeitung aufgetreten ist.

HTTP ist ein zustandsloses Protokoll, daher wird nach jeder Anfrage die Verbindung vom Server wieder abgebaut. Für eine Zuordnung eines Clients muss dieser eine Session-ID mitsenden welche normalerweise im Header enthalten ist.

#### 3.1.2 REST

#### 3.1.3 JSON

#### 3.1.4 AJAX

#### 3.1.5 Objectiv J

Objective J ist eine Programmiersprache welche sich von der Syntax stark an Objective C anlehnt. Sie ist eine Erweiterung oder Obermenge von Javascript und wird von einem in Javascript geschriebenen Interpreter abgearbeitet. In Javascript können Objekte durch Prototyping erstellt werden, das Konzept von Klassen wird aber nicht unterstützt. Obj J bietet zusätzlich zu den nativen JS Objekten die Definition von Klassen inklusive Vererbung und die Generierung von Objekten daraus. Obwohl es die Sprache erlaubt für Variablen, Methodenparameter und Rückgaben einen Datentyp zu definieren, werden diese aufgrund von schwacher Typisierung vom Interpreter nicht auf ihre Einhaltung überprüft. In der aktuellen Version wird die Übergabe von Referenzen als Parameter ähnlich einem Pointer in C unterstützt. [?]

#### 3.1.6 Cappuccino

Bei Cappuccino handelt es sich um ein Web Application Framework zur Erstellung komplexer Benutzeroberflächen in Javascript und Objectiv J. Das Framework lehnt

sich vom Aussehen als auch von den Komponentennamen sehr stark an das GUI-Framework Cocoa von Apple an. GUI-Elemente werden als Objekte erstellt welche von einer View-Klasse erben und innerhalb von anderen Views positioniert werden können. Das Interface wird von einem HTML5 fähigen Browser gerendert wobei für dessen Erstellung keinerlei HTML oder CSS Kenntnisse notwendig sind.

## **3.2 Kommunikation mit KRESHMOI**

### **3.2.1 Query nach Bildern**

### **3.2.2 Laden der Bilder**

## **3.3 Architektur und Komponenten**

### **3.3.1 Domänen Model**

### **3.3.2 Architektur und Aufteilung in Komponenten**

### **3.3.3 2DView**

### **3.3.4 Kommunikations Module**

## **3.4 Usability**

### **3.4.1 Workflow bei der Befundung**

## 4 Results

### 4.1 Geschwindigkeit

### 4.2 Usability

## 5 Conclusion