Online Medical Imaging Platform

Willinger Christin

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

August 5, 2014

Supervisor: Robert Sablatnig

Contents

1 Introduction

1.1 Motivation

1.1.1 Interface für Kreshmoi

Das Ziel von KHRESMOI ist das Durchsuchen und der Zugang zu medizinischen Informationen für verschiedene Benutzergruppen mit unterschiedlichem medizinischen Vorwissen. Die Einteilung der Benutzer erfolgt in 3 Kategorien:

- Personen ohne speziellen medizinischen Kentnissen
- Ärzte
- Radiologen

Dazu verknüpft KHRESMOI Daten aus verschiedenen heterogenen Resourcen wie Bildern aus PACS, Bildern und Text aus Publikationen in Journalen oder Daten von Webseiten. Da sich die verschiedenen Resoucen qualitativ sehr stark voneinander unterscheiden können wird einen Bewertung ihrer Glaubwürdigkeit durchgeführt und dem Benutzer angezeigt.
Die Suchanfrage kann in textueller Form oder als Bild-Query sowie als Kombination von beidem gestellt werden. Ein weiteres wichtiges Feature hiebei ist die multilinguale Suche, da die Menge an verfügbaren medizinischen Informationen nicht in alle Sprachen gleich ist. Dies bedeutet dass die Suchanfrage in mehrere Sprachen übersetzt wird und somit auch anderssprachige Quellen durchsucht werden können. Die Zusammenfassungen der Suchergebnisse werden anschließend in die Anfragesprache rückübersetzt wodurch der Benutzer schnell duch die Ergebnissliste navigieren kann.

1.1.2 Interaktion mit einer Bildsuchmaschiene

Ein Teilprojekt von KHRESHMOI ist das Druchsuchen von medizinischen Bilddaten wobei diese in 2D, 3D oder 4D (Video) vorliegen können. Um eine Suchanfrage auf ein Bild stellen zu können müssen an einem Referenz-Bild Bereiche eingezeichnet werden welche dann die Anfrage formen. Aus der Textur eines markierten Bereiches wird ein Feature-Vektor extrahiert mit dem anschließend eine Datenbank von zuvor indizierten Bildern durchsucht wird.
Ein Frontend einer Bildsuchmaschiene muss daher sowohl Tools zum markieren von interessanten Bereichen, als auch die Funktionalität zur verfnünftigen Berachtung der Bilder bereitstellen.
Diese Arbeit spezialisiert sich auf das Durchsuchen von radioloischen Aufnahmen in 2D und 3D welche in einem PACS (Picture Archiving and Communication Systems) abgelegt sind. Da in einem Krankenhaus täglich große Mengen an Daten durch radionlogische Aufnahmen produziert werden, bietet eine effizientes Durchsuchen dieser die möglichkeit Sie für Ausbildung und Forschug wieder zu verwenden.
Dazu muss das User-Interface die grundliegenden Funktionen eines Betrachtungs-Tools für Röngten- und Computertomographie-Aufnahmen zur verfüfung Stellen:

- Zoom
- Schnelles anpassen von Kontrast und Helligkeit
- Navigation durch die Schnitte eins 3D-Körpers in einer Schnittachse

1.1.3 Hardware und OS Abstraktion

1.2 Pflichtenheft

1.3 Möglichkeiten zur Umsetzung

1.3.1 JavaApplet

1.3.2 HTML5

2 Related Work

2.1 Funktionsumfang von Betrachtungstools

2.1.1 Beispiel Osirix

3 Methodology

3.1 Verwendetet 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 zwischen zwei Kommunikationspartner findet in der From von Nachrichten statt, wobei der Client eine Anfrage an einen Server stellt und dieser die Anfrage bearbeitet und eine Antwort retuniert.
Eine Nachricht setzt sich aus einem Header und einen 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

REST ist im eigentlichen Sin mehr ein Architekturstiel als ein Protokoll welcher mit HTTP umgesetzt wird. Die Idee von REST ist dass eine URL genau eine Resource auf einem Server addressiert, wobei eine Resource eine statische Datei oder das ergebniss einer aktion auf dem Server sein kann.
Der Architekturstiel ist lässt sich durch fünf Prizipien zusammenfassen:

Resource mit eindeutiger Identifikation:
Jede Resrouce wird durch eine URI (Uniform Resouce Identifier) weltweit eindeutig identifiziert. Diese addressiert unter anderem den Server auf den sich die resource befindet sowie Resource auf dem Server selbst.

Hypermedia
Verknüfungen zu anderen Entitäten und werden als Links auf die jeweiligen Resourcen dargestellt. Weiters kann die Steuertung des Applikationszustandes durch Links auf weiter Aktionen durch Hypermedia umgesetzt werden.

Standard-Opperationen
Es gibt ein definiertes Interface welches von jeder Resource zur verfügung gestellt werden muss. Dieses umfasst einen relativ kleinen Satz von Opperationen welche auf die Resource ausgeführt werden können.

Unterschiedliche repräsentation der Resourcen
Die Resourcen können unterschiedliche Darstellungsformen haben. Ein Client kann also eine Resource in einem bestimmten Format (z.B.: XML, HTML, JSON) anfordern sofern diese Darstellung vom Server unterstüzt wird. In HTTP wird die gewünschte Dartstellung im Header angegeben.

Zustandslose Kommunikation
Der Server hällt keine Zustandsinformationen über den Client welcher über die Dauer eines Requests hinaus geht. Daher muss der Zustand einer Anwendung entweder am Client liegen oder vom Server in eine Resource umgewandelt werden.

3.1.3 JSON

3.1.4 AJAX

AJAX (Asynchronous JavaScript and XML) ermöglicht es einer Web Anwendung kleinere Mengen von Daten nachzuladen und damit Teile der Webseite zu ändern, statt bei jeder Aktion die Webseite neu zu laden. Benötigt die Web Appliketion Daten vom Server wird

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 unterstüzt. 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 eine Datentyp zu definieren, werden diese aufgrund von schwacher Typisierung vom Interpreter nicht auf ihre Einhaltung überprüft. In der aktullen 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 für Objectiv J und Javascript, welches haupsächlich der Erstellung komplexer Benutzeroberflächen dient. Das Framework lehnt sich sowol vom Aussehen als auch von der Benennung der Komponenten 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 kentnisse 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