Bachelor-Thesis

Analyse und Erweiterung einer Software zur statistischen Auswertung von Daten aus einem Krebsregister

Studiengang Medizinische Informatik der Hochschule Heilbronn und der Universität Heidelberg.

Vorgelegt von: Tim Happel
Geboren: 03.05.1988
Studienbereich: Medizinische Informatik Bachelor (MIB)
Vorgelegt bei: Prof. Dr. Martin Haag
Korreferent: Prof. Dr. Wendelin Schramm
Eingereicht am: 28.03.2013

Inhaltsverzeichnis

1. Einleitung....3
1.1 Gegenstand und Motivation....3
1.2 Zielsetzung....3
2. Grundlagen....4
2.1 Analyse des Online Clinical Data Mining (OCDM)....4
2.2 R - The R Project for Statistical Computing....7
3. Anforderungsanalyse....8
3.1 Use-Case Diagramm....8
3.2 OP-Art....9
3.3 R-Klassifikation....10
3.4 UICC-Stadien....11
3.5 ECOG-Performance....12
3.6 Symptomatik bei Erstdiagnose....13
3.7 Chemotherapie-Schema....14
3.8 Chemotherapie-Substanz....15
4. Implementierung....16
4.1 Klassendiagramme....16
4.2 Sequenzdiagramme....21
5. Systemtest....24
5.1 Einleitung
....24
5.2 Testumgebung....24
5.3 Abnahmekriterien....25
5.4 Testabschnitt - Pankreaskarzinom....25
6. Abschluss....30
6.1 Diskussion....30
6.2 Ausblick....30
Literaturverzeichnis....31
Abbildungsverzeichnis....32

1. Einleitung

1.1 Gegenstand und Motivation

Der Zuwachs medizinischen Wissens ist gewaltig, man spricht von einer Verdopplung in unter 10 Jahren[01], dies macht die Analyse historischer Daten zwingend erforderlich. Um dieses Wissen zu beherrschen werden Datenbanken benötigt, in denen man Krankheits- und Therapieverläufe ablegen kann, um diese dann anschließend unter unterschiedlichsten Anforderungen zu analysieren. Im Falle der Krebstherapie gibt es hierfür sogenannte Krebsregister. Hier werden anonymisierte Daten der Patienten gespeichert, wie zum Beispiel der Verlauf der Krankheit und die Therapie. Ziel ist es, dass alle Krebsfälle in irgendeinem Krebsregister gehalten werden. Dies soll zu einer Verbesserung der medizinischen Leistung am Patienten, sowie der Sicherstellung, dass Patienten überall mit neuesten Therapietechniken und Leistungen versorgt werden, führen. Leider werden die Informationen in den Krebsregistern viel zu wenig genutzt, da viele Ärzte und Wissenschaftler oft nicht Ausreichend vertiefte Kenntnisse in Informationstechnologie und/oder Statistik haben. Diese Tatsache macht es für sie schwierig die riesigen Datenmengen, die vorhanden sind, richtig zu analysieren. Um dieses Problem zu beheben kann man nun in regelmäßigen Abständen einen Statistiker beauftragen, der solche Analysen durchführt. Oft haben Ärzte aber statistisch wenig anspruchsvolle Anfragen oder es fällt ihnen erst auf, nachdem der Statistiker wieder gegangen ist. Solange es also eine solche Schwierigkeit darstellt, die Daten richtig zu analysieren, sind die Krebsregister ein Datenfriedhof, deren riesiges Potential nicht ausgenutzt wird. Dieser Zustand ist sowohl für Patienten, als auch für die Ärzte, nicht zufriedenstellend und bedarf dringend einer Änderung.

1.2 Zielsetzung

Die Ziele sind:
1. Eine Analyse einer Software zur statistischen Auswertung von Daten aus einem Krebsregister (OCDM-Software), sowohl im Bezug auf ihre Software-Architektur, als auch auf ihre Funktion.
2. Das Erstellen einer Anforderungsanalyse, welche die Erweiterungen beschreibt, die an der oben erwähnten OCDM-Software vorzunehmen sind.
3. Die Umsetzung dieser Anforderungsanalyse in die bestehende Anwendung.
4. Ein abschließender Systemtest der Anwendung, um einen reibungslosen Ablauf im Klinikalltag zu gewährleisten.

Ziel der Erweiterung ist es, den Ärzten die Analyse der gesammelten Patientendaten zum Pankreaskarzinom zu vereinfachen und somit die medizinische Betreuung in der Klinik zu verbessern.

2. Grundlagen

2.1 Analyse des Online Clinical Data Mining (OCDM)

Das OCDM ist eine Anwendung, die es dem Anwender erlaubt, ohne große Vorkenntnisse, statistische Anfragen zu verschiedenen Krebsarten zu stellen. Selbstverständlich wird heutzutage jeder Krankheitsfall im Krankenhaus sehr genau dokumentiert. Diese Dokumentation ist aber nicht nur zu rechtlichen Zwecken da, sondern auch, damit diese Daten Aufschluss über die Verläufe von Krankheiten geben können oder die Auswirkungen einzelner Faktoren auf den Verlauf einer Krankheit aufzeigen. Im Falle des OCDM liegt dabei das Augenmerk ausschließlich auf Krebserkrankungen. Die Patientendaten werden zur Verwendung im OCDM in eine extra Datenbank geladen und stehen dann zur Verfügung. Die Datenbank wurde so angelegt, dass die Daten in der Form vorliegen, wie sie zur Verwendung in statistischen Verfahren gebraucht werden [02]. Derzeit ist die Analyse fünf verschiedener Karzinomarten möglich, nämlich des Mamma-, Bronchial-, Rektum-, Kolon- und Prostatakarzinoms. Das OCDM bietet hierfür die Möglichkeit, bis zu 4 Profile(Kliniken) im Bezug auf verschiedene Merkmale und deren Ausprägungen, miteinander zu vergleichen. Alternativ kann man per Cox-Regression Merkmale miteinander vergleichen, um deren Auswirkung auf das Überleben der Patienten sichtbar zu machen. Merkmale können zum Beispiel die ECOG-Performance oder die PNM-Tumorklassifikation sein. Hat man nun die Anfrage ausgewählt, was mit Hilfe von einfachen Dropdown-Menüs geschieht, erzeugt die Anwendung Grafiken, die einem das Ergebnis anschaulich zeigen. Neben Balkendiagrammen, die, je nach Auswahl, absolute oder relative Häufigkeiten der Merkmale anzeigen, ist auch die Generierung der Kaplan-Meier-Überlebenskurve und der Hazard-Kurve möglich. Die Einfachheit der Auswahl dieser Merkmale und deren Darstellung erlaubt es, die Anwendung auch ohne große technische oder statistische Grundkenntnisse zu bedienen.

Im Einleitungskapitel wurde bereits vom ständig größer werdenden medizinischen Wissen und dessen Abspeicherung, im Falle des Karzinoms in den sogenannten Krebsregistern, gesprochen. Abgespeichert werden hierbei anonymisierte Patientendaten mit allem, was für eine Analyse wichtig sein könnte. Dazu gehören zum Beispiel Alter des Patienten, das Datum der Erstdiagnose oder die Tatsache ob er noch lebt. Diese Daten sollen durch Anwendungen, wie dem OCDM, einem sinnvollen Zweck zugeführt werden. Klar ist, dass die Anwendung keine genauere statistische Analyse ersetzt oder gar Grundlagen zu medizinischen Studien schaffen kann. Aber das OCDM kann einem eine Richtung weisen, da man sehr einfach ein Gefühl dafür bekommt, welche Tendenzen sich zeigen, um dann gezielt einen Statistiker zu beauftragen, der in dieser Richtung weitere Nachforschungen anzustellen kann. Die Suche nach neuen Beweisen und die Formulierung neuer Hypothesen wird von den Entwicklern des OCDM als Hauptgrund für die Entwicklung des Programms angegeben [03].
Das OCDM soll also dem Problem Abhilfe schaffen, dass Statistiker nur sehr selten in die Kliniken kommen und es manchmal für die Ärzte schwierig ist zu wissen in welche Richtung sie genauere Untersuchen machen wollen. Weitere Informationen sowie Kontaktdaten finden sie auf der offiziellen Homepage des OCDM [04].

Das OCDM ist eine, mit Java realisierte, Webanwendung. Es besteht aus 3 großen Teilen.

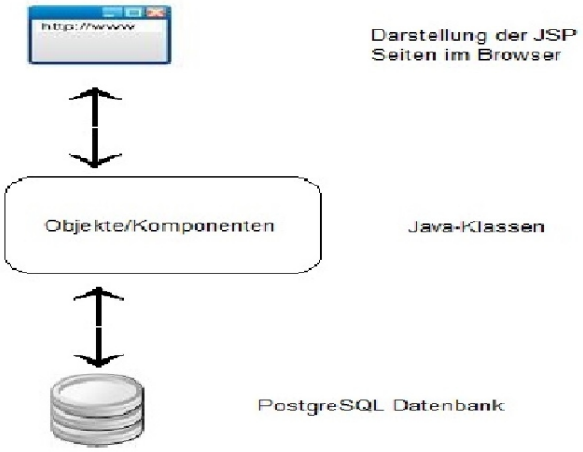

Abbildung 1: Grober Aufbau und Kommunikation der Ebenen.

Die Daten werden, wie in Abbildung 1 dargestellt, in einer PostgreSQL Datenbank gehalten. In der Datenbank befinden sich die Tabellen zu den verschiedenen Karzinomarten und den zugehörigen Merkmalen. Außerdem die, nach Geschlechtern getrennten, Sterbetafeln, welche die Sterblichkeit Verhältnisse der deutschen Bevölkerung, laut statistischem Bundesamt, beinhalten . Diese Daten werden benötigt, um die Überlebenskurve und Hazard-Kurve zu berechnen. Neben den Tabellen befinden sich in der Datenbank Funktionen, welche das Auslesen der Daten aus den verschiedenen Tabellen übernehmen und diese Daten der Geschäftslogik zur Verfügung stellen. Daten werden im OCDM nur aus der Datenbank ausgelesen, Schreibvorgänge gibt es nicht.
Die Geschäftslogik teilt sich in 7 Packages auf.

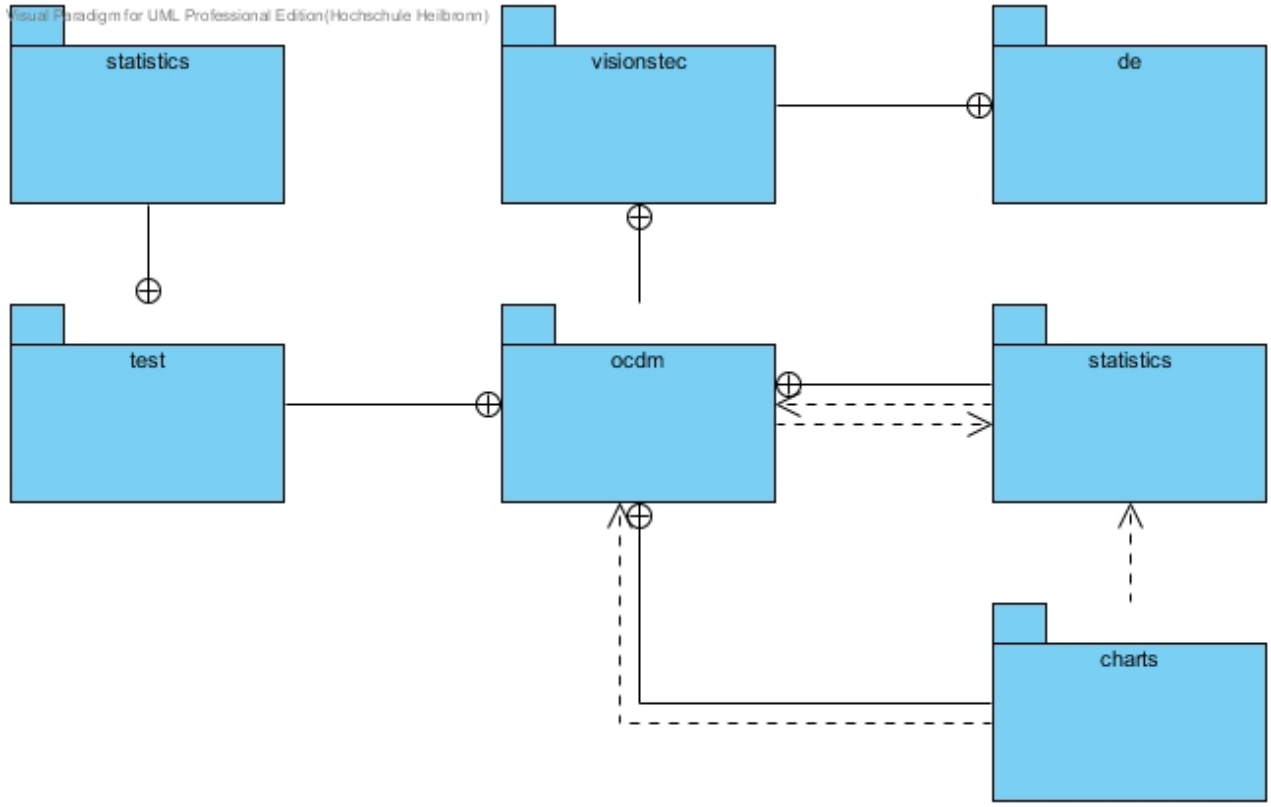

Abbildung 2: Paketdiagramm des OCDM

Die Packages „de“, „test“ und „visiontec“ haben lediglich eine strukturierende Aufgabe.
Wie im Paketdiagramm zu erkennen bildet das Package „ocdm“ das Herzstück, es enthält hauptsächlich Utility-Klassen, welche die Weiterleitung der .jsp Seiten oder das Erstellen der verschiedenen Diagramme managen. Das Package „charts“ enthält Klassen zum Erstellen der verschiedenen Diagramme. Diese werden mit Hilfe der Bibliothek „JFreeChart“ dargestellt.
Das Package „statistics“ enthält die Methoden zur Berechnung der verschiedenen Statistiken. Zur Berechnung der Statistiken wird das Programm R verwendet, welches im Kapitel 2.2 genauer erklärt wird. Das Package „test.statistics“ enthält nur einen Test für die Klasse, die den Chi-Quadrat-Test, durchführt und ist somit relativ unspektakulär.
Die Präsentation der Diagramme erfolgt, wie in Abbildung 1 zu sehen, im Browser und ist in der Anwendung über JSF 1.2 und JSP-Seiten realisiert. In Abbildung 3 ist die Oberfläche dargestellt, in welcher man sich die verschiedenen Diagramme anzeigen lassen kann, hier im Beispiel zum Mammakarzinom.

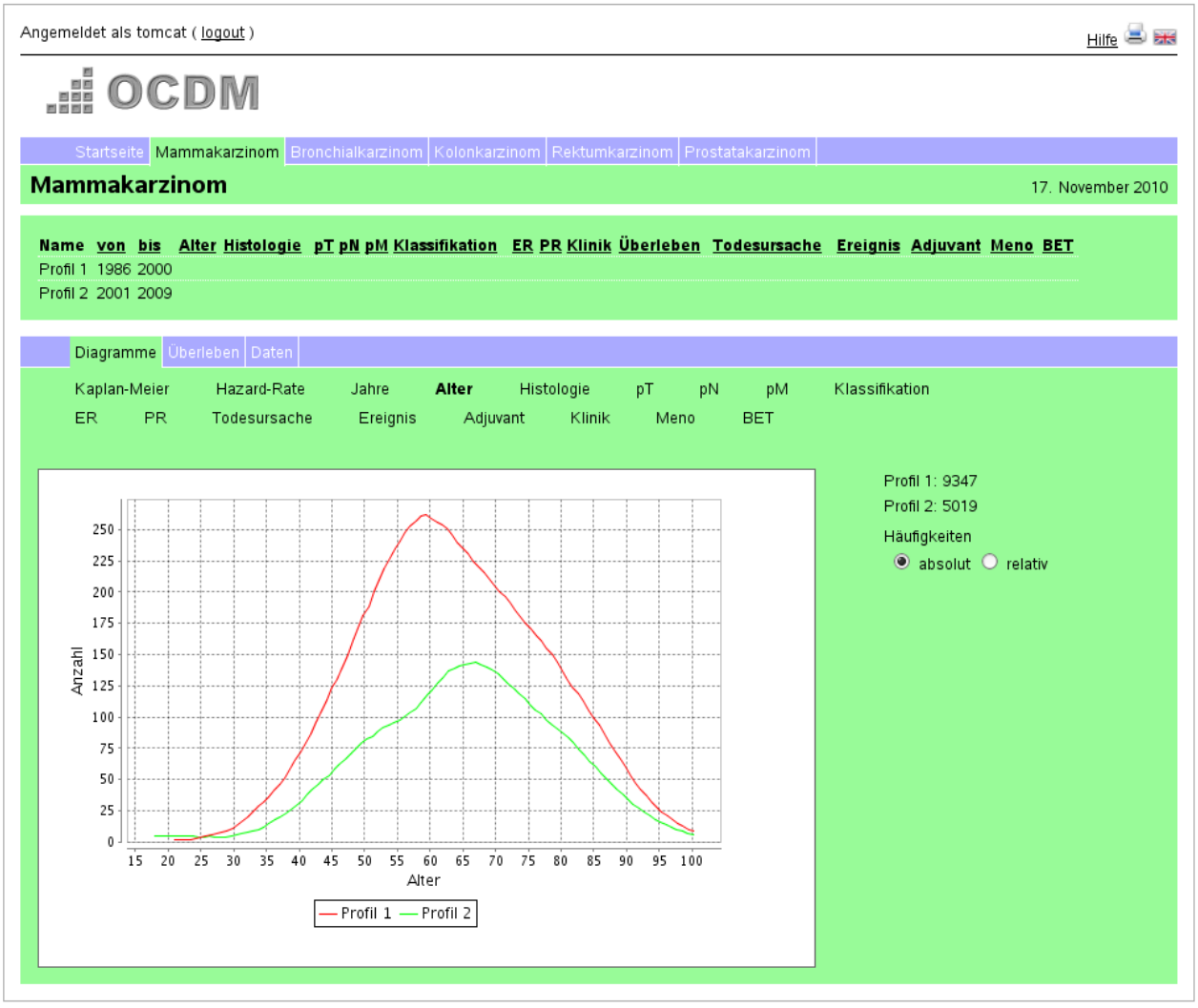

Abbildung 3: Das Frontend des OCDM, mit einem Beispiel Diagramm

2.2 R - *The R Project for Statistical Computing*

R ist eine Freie Software, welche eine Vielzahl statistischer Berechnungen, wie statistische Tests, lineare und nicht-lineare Modellierung und vieles mehr, sowie grafische Darstellung dieser Berechnungen, anbietet. R ist unter Statistikern eines der am häufigsten eingesetzten Programme zur statistischen Analyse [05]. Obwohl R zunächst in einer Kommandozeilenumgebung läuft, kann es durch Pakete um eine grafische Benutzeroberfläche erweitert werden, wie zum Beispiel die Java-basierte Benutzeroberfläche JGR („Java GUI for R“), die einem unter anderem auch gewisse Syntaxüberprüfung bietet. R kann sehr leicht durch eine Vielzahl verfügbarer Pakete erweitert werden, welche sehr einfach über das Zentrale Archiv, Comprehensive R Archive Network(CRAN), installiert werden können. Dazu gehört zum Beispiel auch, die im OCDM verwendete Bibliothek 'splines', welche zur Berechnung der Kaplan-Meier Überlebenskurve verwendet wird. Weitere Informationen zu R, sowie die Software selbst finden sie auf der offiziellen Homepage des Projektes [06].
Das folgende Beispiel soll die Syntax und Arbeitsweise von R ein wenig verdeutlichen.
Gezeigt wird wie Variablen zugewiesen werden und Methoden Aufrufe funktionieren.

```
# x wird als numerischer Vektor durch den Zuweisungsoperator "<-" definiert:
> x <- c (1, 2, 3, 4, 5, 6)
> mean(x)            # Berechnet das Arithmetische Mittel des Vektors x
[1] 15.16667
```

Hier nun ein Beispiel, wie eine Ausgabe auf der Konsole von R aussehen kann:

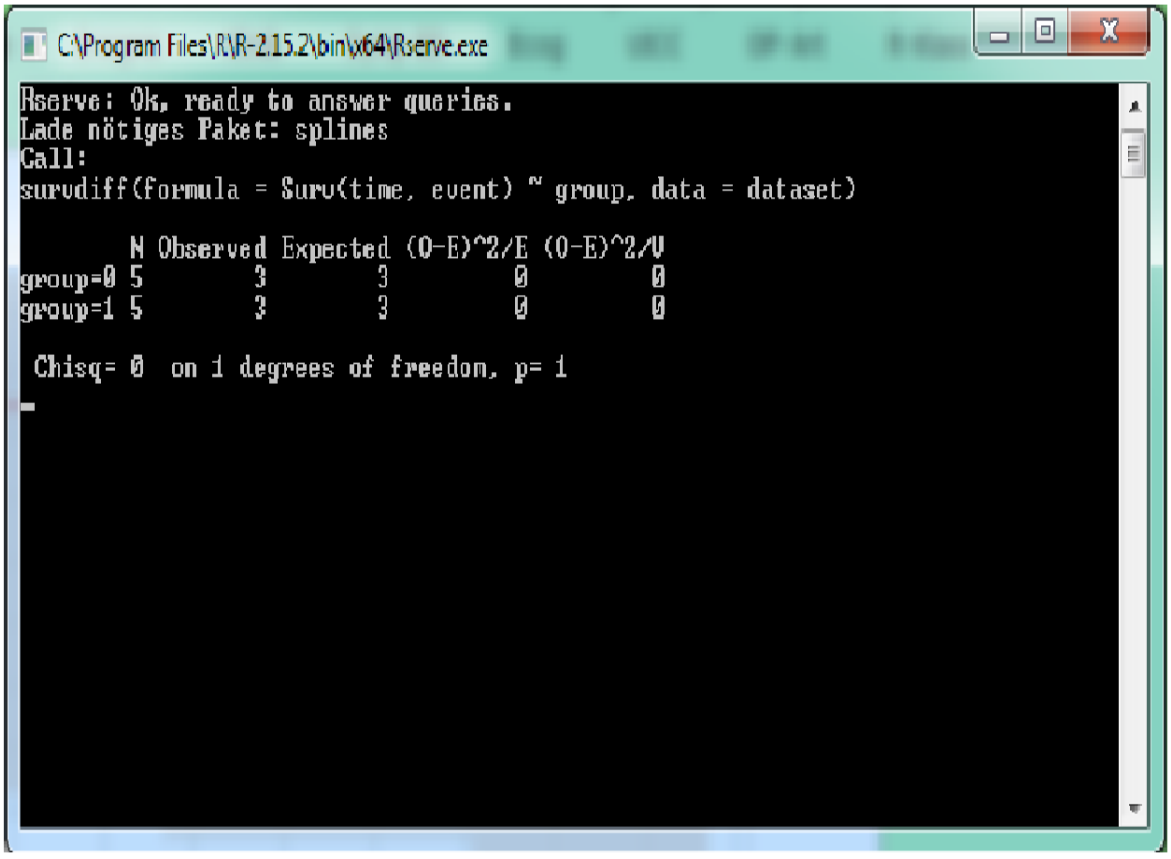

Abbildung 4: Konsolen Ausgabe von R, nach Berechnung einer Kaplan-Meier Überlebenskurve

3. Anforderungsanalyse

Dieses Kapitel beschreibt die Aufgabenstellung, wie sie mit den Ärzten der SLK-Kliniken abgesprochen wurde. Sie ist mit Hilfe von Use-Case Diagrammen aufgelistet und analysiert.

3.1 Use-Case Diagramm

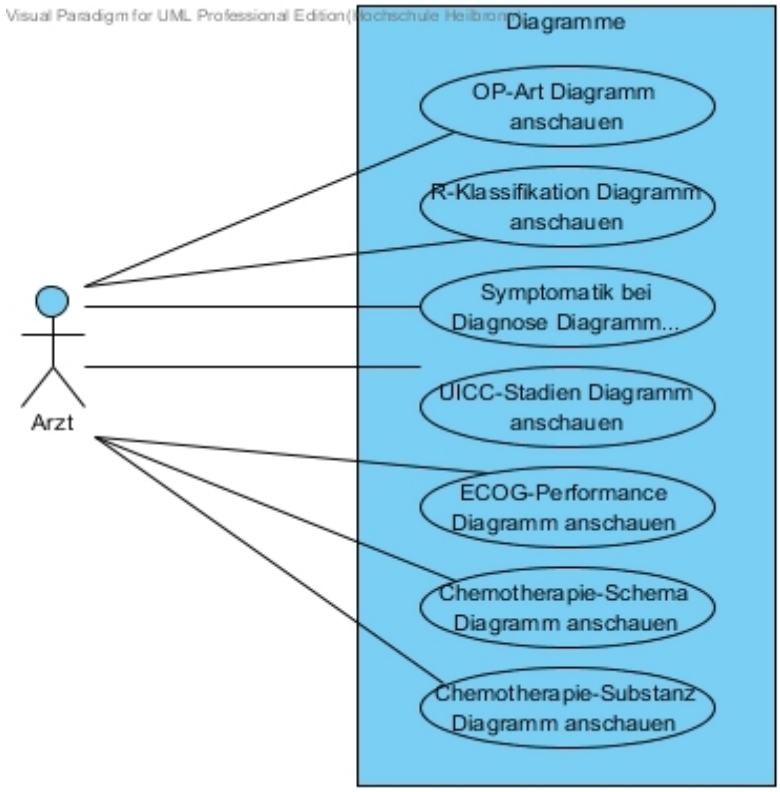

Abbildung 5: Use-Case Diagramm

3.2 OP-Art

Name des Use-Cases:
OP-Art Diagramm anschauen
Subsystem:
Diagramme
Aktoren:
Arzt
Kontext und Vorbedingungen:
-
Normalablauf:

1. Der Arzt wählt die Art der Statistik aus folgender Liste aus:
   a) Subgruppen
   b) Ähnliche Fälle
   c) Cox-Regression
2. Der Arzt wählt ein Anfangs- und Endjahr aus, welche den Zeitraum der zu berücksichtigten Patienten beschreibt.
3. Der Arzt wählt die Art der OP aus die er untersuchen will:
   a) Alle
   b) totale Duodenopankreatektomie
   c) partielle Duodenopankreatektomie
   d) Pankreaslinksresektion
   e) explorative Laparotomie ohne Pankreasresektion
4. Der Arzt wählt aus, ob die Grafik absolute oder relative Werte anzeigen soll.
5. Es wird eine Grafik generiert, welche die absoluten/relativen Zahlen, je nach Auswahl aus Punkt „4.“, der Patienten mit der ausgewählten OP-Art, anzeigt.

Normalergebnis:
Die generierte Grafik wird angezeigt.
Alternativablauf:
Falls keine Daten vorhanden sind, wird kein Graph angezeigt.
Nicht-Funktionale Anforderungen:
-

3.3 R-Klassifikation

Name des Use-Cases:
R-Klassifikation Diagramm anschauen
Subsystem:
Diagramme
Aktoren:
Arzt
Kontext und Vorbedingungen:
-
Normalablauf:

1. Der Arzt wählt die Art der Statistik aus folgender Liste aus:
   a) Subgruppen
   b) Ähnliche Fälle
   c) Cox-Regression
2. Der Arzt wählt ein Anfangs- und Endjahr aus, welche den Zeitraum der zu berücksichtigten Patienten beschreibt.
3. Der Arzt wählt die R-Klassifikation aus die er untersuchen will:
   a) Alle
   b) R0
   c) R1
   d) R2
   e) RX
4. Der Arzt wählt aus, ob die Grafik absolute oder relative Werte anzeigen soll.
5. Es wird eine Grafik generiert, welche die absoluten/relativen Zahlen, je nach Auswahl aus Punkt „4.“, der Patienten mit der ausgewählten R-Klassifikation, anzeigt.

Normalergebnis:
Die generierte Grafik wird angezeigt.
Alternativablauf:
Falls keine Daten vorhanden sind, wird kein Graph angezeigt.
Nicht-Funktionale Anforderungen:
-

3.4 UICC-Stadien

Name des Use-Cases:
UICC-Stadien Diagramm anschauen
Subsystem:
Diagramme
Aktoren:
Arzt
Kontext und Vorbedingungen:
-
Normalablauf:

1. Der Arzt wählt die Art der Statistik aus folgender Liste aus:
   a) Subgruppen
   b) Ähnliche Fälle
   c) Cox-Regression
2. Der Arzt wählt ein Anfangs- und Endjahr aus, welche den Zeitraum der zu berücksichtigten Patienten beschreibt.
3. Der Arzt wählt das UICC-Stadium aus das er untersuchen will:
   a) Alle
   b) 0
   c) IA
   d) IB
   e) IIA
   f) IIB
   g) III
   h) IV
4. Der Arzt wählt aus, ob die Grafik absolute oder relative Werte anzeigen soll.
5. Es wird eine Grafik generiert, welche die absoluten/relativen Zahlen, je nach Auswahl aus Punkt „4.“, der Patienten mit dem ausgewählten UICC Stadium, anzeigt.

Normalergebnis:
Die generierte Grafik wird angezeigt.
Alternativablauf:
Falls keine Daten vorhanden sind, wird kein Graph angezeigt.
Nicht-Funktionale Anforderungen:
-

3.5 ECOG-Performance

Name des Use-Cases:
ECOG-Performance Diagramm anschauen
Subsystem:
Diagramme
Aktoren:
Arzt
Kontext und Vorbedingungen:
-
Normalablauf:

1. Der Arzt wählt die Art der Statistik aus folgender Liste aus:
   a) Subgruppen
   b) Ähnliche Fälle
   c) Cox-Regression
2. Der Arzt wählt ein Anfangs- und Endjahr aus, welche den Zeitraum der zu berücksichtigten Patienten beschreibt.
3. Der Arzt wählt die ECOG-Performance aus die er untersuchen will:
   a) Alle
   b) ECOG: 0
   c) ECOG: 1
   d) ECOG: 2
   e) ECOG: 3
   f) ECOG: 4
4. Der Arzt wählt aus, ob die Grafik absolute oder relative Werte anzeigen soll.
5. Es wird eine Grafik generiert, welche die absoluten/relativen Zahlen, je nach Auswahl aus Punkt „4.“, der Patienten mit der ausgewählten ECOG-Performance, anzeigt.

Normalergebnis:
Die generierte Grafik wird angezeigt.
Alternativablauf:
Falls keine Daten vorhanden sind, wird kein Graph angezeigt.
Nicht-Funktionale Anforderungen:
-

3.6 Symptomatik bei Erstdiagnose

Name des Use-Cases:
Symptomatik bei Diagnose Diagramm anschauen
Subsystem:
Diagramme
Aktoren:
Arzt
Kontext und Vorbedingungen:
-
Normalablauf:

1. Der Arzt wählt die Art der Statistik aus folgender Liste aus:
   a) Subgruppen
   b) Ähnliche Fälle
   c) Cox-Regression
2. Der Arzt wählt ein Anfangs- und Endjahr aus, welche den Zeitraum der zu berücksichtigten Patienten beschreibt.
3. Der Arzt wählt das Vorhandensein einer Symptomatik bei Diagnose aus das er untersuchen will:
   a) Alle
   b) Ja
   c) Nein
4. Der Arzt wählt aus, ob die Grafik absolute oder relative Werte anzeigen soll.
5. Es wird eine Grafik generiert, welche die absoluten/relativen Zahlen, je nach Auswahl aus Punkt „4.“, der Patienten mit der ausgewählten Symptomatik bei Diagnose, anzeigt.

Normalergebnis:
Die generierte Grafik wird angezeigt.
Alternativablauf:
Falls keine Daten vorhanden sind, wird kein Graph angezeigt.
Nicht-Funktionale Anforderungen:
-

3.7 Chemotherapie-Schema

Name des Use-Cases:
Chemotherapie-Schema Diagramm anschauen
Subsystem:
Diagramme
Aktoren:
Arzt
Kontext und Vorbedingungen:
-
Normalablauf:

1. Der Arzt wählt die Art der Statistik aus folgender Liste aus:
   a) Subgruppen
   b) Ähnliche Fälle
   c) Cox-Regression
2. Der Arzt wählt ein Anfangs- und Endjahr aus, welche den Zeitraum der zu berücksichtigten Patienten beschreibt.
3. Der Arzt wählt das Chemotherapie-Schema aus das er untersuchen will:
   a) Alle
   b) Adjuvant
   c) Neoadjuvant
   d) Palliativ
4. Der Arzt wählt aus, ob die Grafik absolute oder relative Werte anzeigen soll.
5. Es wird eine Grafik generiert, welche die absoluten/relativen Zahlen, je nach Auswahl aus Punkt „4.“, der Patienten mit dem ausgewählten Chemotherapie Schema, anzeigt.

Normalergebnis:
Die generierte Grafik wird angezeigt.
Alternativablauf:
Falls keine Daten vorhanden sind, wird kein Graph angezeigt.
Nicht-Funktionale Anforderungen:
-

3.8 Chemotherapie-Substanz

Name des Use-Cases:
Chemotherapie-Substanz Diagramm anschauen
Subsystem:
Diagramme
Aktoren:
Arzt
Kontext und Vorbedingungen:
-
Normalablauf:

1. Der Arzt wählt die Art der Statistik aus folgender Liste aus:
   a) Subgruppen
   b) Ähnliche Fälle
   c) Cox-Regression
2. Der Arzt wählt ein Anfangs- und Endjahr aus, welche den Zeitraum der zu berücksichtigten Patienten beschreibt.
3. Der Arzt wählt die, in der Chemotherapie, verwendete Substanz aus, die er untersuchen will:
   a) Alle
   b) Gemicitabin
   c) Gemicitabin/Capecitabin
   d) Gemicitabin/Erlotibin
   e) FOLFOX 6
   f) Kombinierte Radiochemotherapie
   g) 5-FU
   h) BAGPAC
4. Der Arzt wählt aus ob, die Grafik absolute oder relative Werte anzeigen soll.
5. Es wird eine Grafik generiert, welche die absoluten/relativen Zahlen, je nach Auswahl aus Punkt „4.“, der Patienten mit der ausgewählten Chemotherapie Substanz, anzeigt.

Normalergebnis:
Die generierte Grafik wird angezeigt.
Alternativablauf:
Falls keine Daten vorhanden sind, wird kein Graph angezeigt.
Nicht-Funktionale Anforderungen:
-

4. Implementierung

Das Nachfolgende Kapitel beschreibt die Änderungen die nötig waren um die, im vorangegangenen Kapitel beschriebenen, Use-Cases zu implementieren. Hauptsächlich wurden dazu Klassen erstellt, welche es ermöglichen, die in Kapitel 3 beschriebenen, Merkmale als Diagramme darzustellen, diese werden in Kapitel 4.1.1 näher beschrieben. Außerdem wurden Methoden eingefügt, welche das Erzeugen der neuen Diagramme delegieren, diese werden in Kapitel 4.1.2 näher beschrieben.

4.1 Klassendiagramme

4.1.1 Charts

Abbildung 6 zeigt einen Teil des Klassendiagramms des OCDM, alle Klassen befinden sich im Package „charts“. Das Augenmerk liegt hierbei auf den, für das neue Modul „Pankreaskarzinom“ erstellten Klassen, welche die Diagramme für die neuen Merkmale erzeugen. Die Oberklassen „Chart“, „BarChart“, „OCDMBarChart“ und „Profile“ waren schon vorhanden. Dort befinden sich auch die “Profile“ Oberklasse und die Abgeleitete Klasse „ProfilePancreas“, welche die zum neuen Modul gehörigen Daten halten. Getter und Setter Methoden wurde aus Gründen der Übersichtlichkeit weggelassen.
Die Oberklasse „Chart“ stellt hauptsächlich die abstrakte Methode „getChart()“ zur Verfügung, welche ein „JFreeChart“ zurück liefert. Das bedeutet, dass eine Implementierung dieser Methode tatsächlich ein Diagramm zeichnet und zwar mit Hilfe der Bibliothek „JFreeChart“. Die Implementierung dieser Methode erfolgt hierbei durch die Klasse „BarChart“, welche ein Barchart erstellt. Andere Diagrammarten sind zum Beispiel das „Linechart“. Zum Erzeugen eines Diagramms sind natürlich einige Informationen notwendig. Man benötigt die Daten, Achsenbeschriftungen, den Namen der Spalte in der Datenbank und die möglichen Ausprägungen. Da diese Informationen natürlich, je nach Diagramm, unterschiedlich sind, sind die Getter-Methoden abstrakt und werden von den Merkmalsklassen implementiert. Als Beispiel zur Verdeutlichung:
Die Klasse „RClassChart“:

```
@Override //Beschreibt den Spaltennamen in der Datenbank
protected String getColumn() {
	return "rclass";
}


@Override //Beschreibt die Ausprägungen, die angenommen werden können
protected String[] getValues() {
	String[] value = new String[4];
	value[0] = "R0";
	value[1] = "R1";
	value[2] = "R2";
	value[3] = "RX";
	return value;
}
```

```
@Override //Setzt die Achsenbeschriftungen, je nachdem ob relative oder absolute Häufigkeiten angezeigt werden sollen
	protected String[] getAxisLabels() {
		String[] value = new String[2];
		value[0] = getLanguage().getMessages().getString("chart_rclass_xtitle");
		if (isRelative()) {
			value[1] = 
getLanguage().getMessages().getString("chart_rclass_ytitle_rel");
		} else {
			value[1] = 
getLanguage().getMessages().getString("chart_rclass_ytitle");
		}
		return value;
	}
```

Die Implementierung „getData()“ der Klasse“OCDMBarChart“, holt sich die Daten aus der Datenbank anhand des Spaltennamens den „RClassChart“ gesetzt hat und berechnet dann das Histogramm. Das Ergebnis wird dann von der Implementierung der Methode „getChart()“ in der Klasse „BarChart“ gezeichnet.

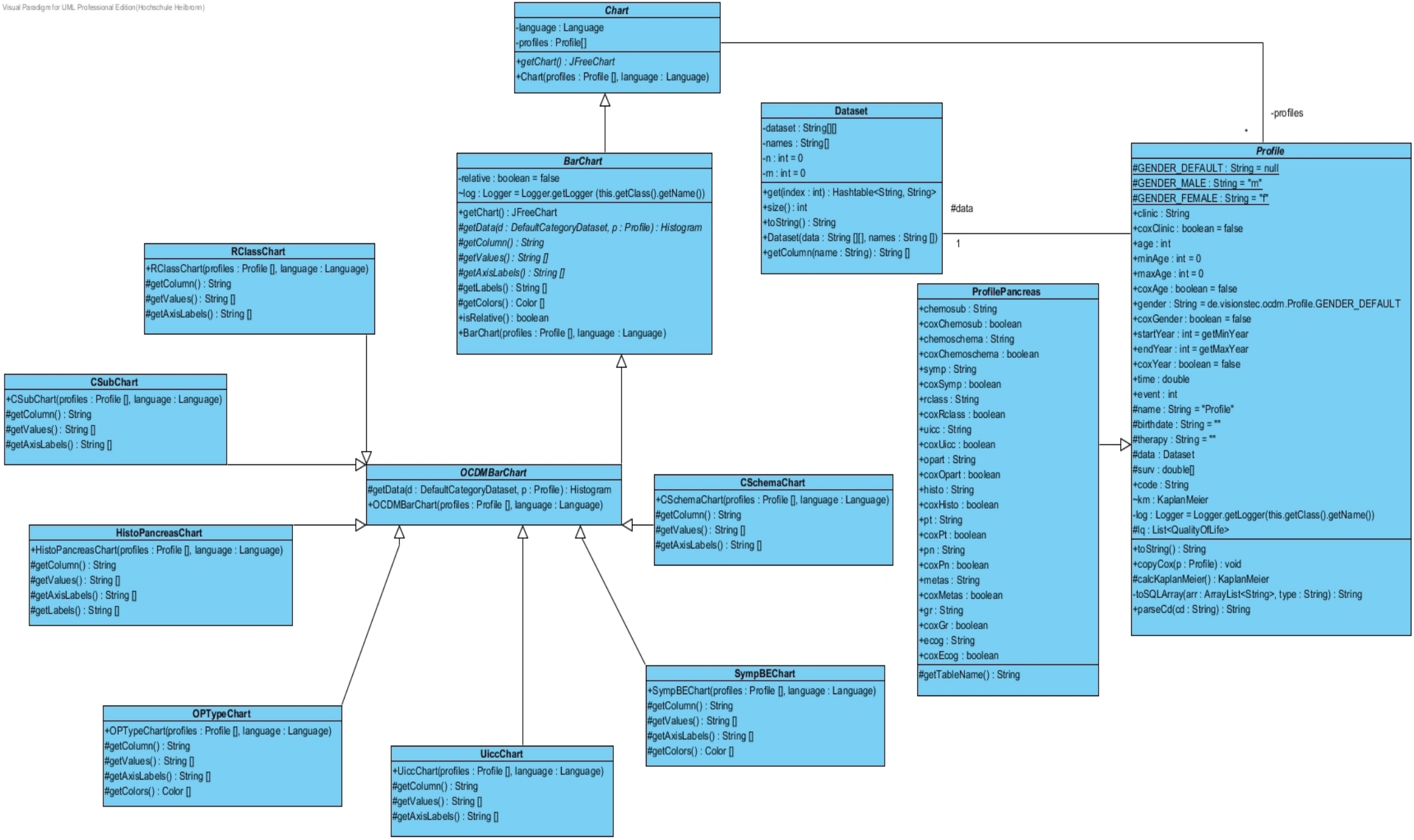

Abbildung 6: Teil des Klassendiagramms mit Augenmerk auf die Charts

4.1.2 Control

Die zwei in Abbildung 7 gezeigten Klassen dienen hauptsächlich dem Delegieren der verschiedenen Aufgaben. Die Klasse „Control“ hält die Namen der Seiten, die angezeigt werden können, sowie die dazugehörigen „Profile“ Klassen. Ein Beispiel einer Profile-Klasse ist in Abbildung 6 zu sehen, die Klasse „ProfilePancreas“. Diese Informationen benötigt „Control“, da sie die Methoden, welche zur Weiterleitung auf die verschiedenen JSP-Seiten dienen, implementiert. Die entsprechenden Methoden sind in der Abbildung 7 gekennzeichnet. Außerdem hat die Klasse Methoden, welche aus den JSP-Seiten aufgerufen werden und das Anzeigen der Diagramme delegieren. Auch diese Methoden sind in Abbildung 7 gekennzeichnet. Da es sich hier um eine Webanwendung handelt und man dabei sehr gut mit HTTP-Requests arbeiten kann, wird das Delegieren der Diagrammaufrufe, durch das Setzen eines Parameters in der URL gelöst. Um das Beispiel aus 4.1.1 aufzugreifen, würde das etwa so aussehen:

```
chartUrl = chartBaseUrl+"?ch=rclass";
```

Das Auslesen des Parameters „ch“ und die Interpretation, welches Diagramm dargestellt werden soll, übernimmt die Klasse „ChartServlet“, auf die gleich noch genauer eingegangen wird. Neben diesen zwei Typen von Methoden zur Änderung der Anzeige, hat die Klasse noch einige Utility-Methoden, die aber nicht weiter von größerem Interesse sind.

Die letzten Methoden der Klasse „Control“ sind, wie in Abbildung 7 beschrieben, Methoden, die zur Lösung der Aufgabenstellung dieser Thesis notwendig waren. Dazu gehören zum einen die Methoden zur Anzeige der neu erstellten Diagramme und zum anderen die Methode zur Weiterleitung auf die neu erstellten JSP-Seiten.

Die Klasse „ChartServlet“ dient wie schon weiter oben erwähnt dazu, den HTTP-Request zum Anzeigen der Diagramme zu verarbeiten und das gewünschte Diagramm zu liefern.

Um diese Aufgabe zu erfüllen erbt die Klasse von „HttpServlet“. Die Methoden „doGet()“ und „doPost()“, sind also die bekannten Methoden, welche vom Http, aufgerufen werden.

Die Methode „getControl“ liefert die Instanz der Klasse „Control“, welche in der aktuellen Session verwendet wird. Die genaue Funktion ist in Kapitel 4.2.1 erklärt.

Wichtig für das Anzeigen der Diagramme sind die übrigen beiden Methoden „parseRequest()“ und „processRequest()“.

„parseRequest()“ übernimmt dabei die oben angedeutete Aufgabe, den in der URL gesetzten, Parameter „ch“ zu interpretieren. Hierfür wird durch eine simple Aneinanderreihung von „if-else“ Blöcken, der Parameter in die entsprechende Konstante umgewandelt, welche in Abbildung 7, in der Klasse ChartServlet, zu sehen ist. Im Beispiel von oben sieht das so aus:

```
else if (ch.equals("rclass")) {
return REQ_RCLASS; }
```

Die Methode „processRequest()“ veranlasst dann das Erstellen des entsprechenden Diagramms. Wie dies im Detail passiert wird in Kapitel 4.2.2 beschrieben.

Abbildung 7: Die Klassen zur Delegation

4.2 Sequenzdiagramme

Dieses Kapitel beschreibt exemplarisch den technischen Ablauf innerhalb des OCDM, um ein besseres Verständnis von der Arbeitsweise des OCDM zu bekommen.

4.2.1 getControl()

Da die Klasse „Control“ zustandsabhängige Daten enthält, ist es nötig, immer die mit der aktuellen Session verknüpfte Instanz der Klasse zu bekommen. Deshalb ist die Instanz der Klasse in der Http Session als Attribut hinterlegt. Man holt sich also zunächst die aktuelle Session, welche im Request hinterlegt ist. Dann ruft man an dieser Session die Methode „getAttribute()“ auf, welcher man den Namen des gewünschten Attributs als Übergabeparameter mitgibt. In diesem Falle den String „control“. Die Http Session liefert einem dann die Instanz der Klasse „Control“, welche aktuell verwendet wird.

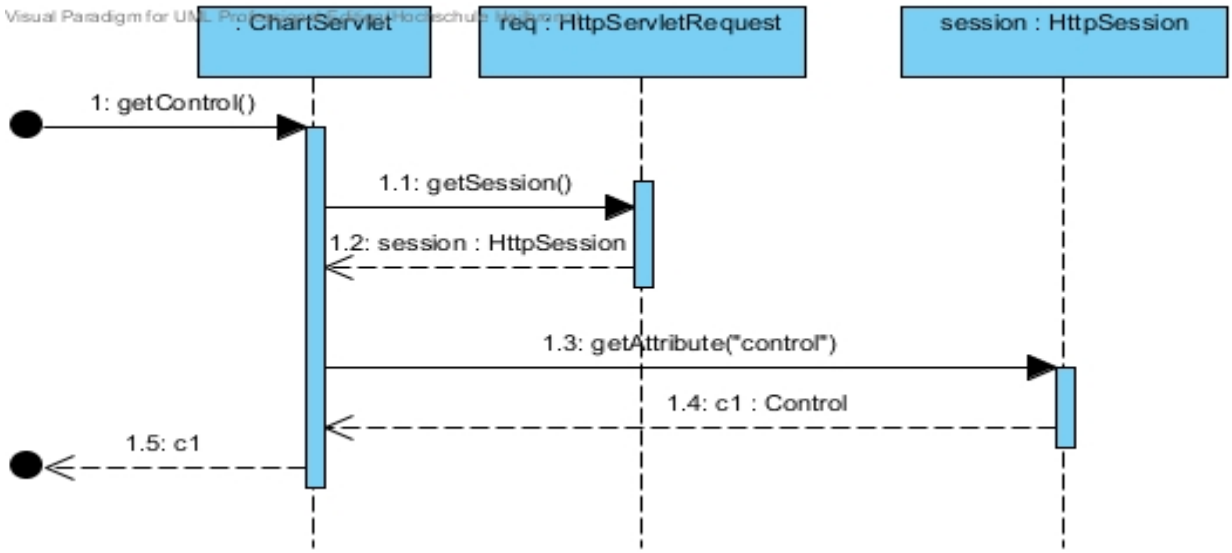

Abbildung 8: Sequenzdiagramm zum Holen des „Control“ Objekts

4.2.2 processRequest

In Kapitel 4.1 wurde am Beispiel der R-Klassifikation gezeigt, wie ein Diagramm erstellt wird. Die Abbildung 9 zeigt die Sequenz wie man dort hinkommt. Als Beispiel dient auch hier wieder die Klasse „RClassChart“.
Zunächst ist zu beachten, dass es sich hier um einen HTTP-Request handelt.
Als erstes wird, die aktuell verwendete, Instanz der Klasse „Control“ geholt. Wie das funktioniert wurde in Kapitel 4.2.1 beschrieben. Der Aufruf „getProfiles()“ der Klasse „Control“, liefert die Anzahl der Profile, die bei der aktuellen Anfrage miteinander verglichen werden sollen.
Nach dieser Größe wird ein Array erstellt, welches so groß ist wie die Anzahl der Profile.
Mit Profilen sind hierbei nicht die verschiedenen Karzinomarten gemeint, sondern die verschiedenen Kliniken.
Der Aufruf „getProfilesList()“ liefert nun eine ArrayList mit den genauen Bezeichnungen dieser Profile. Diese Bezeichnungen werden in der anschließenden Schleife dem vorher erzeugten Array zugewiesen. Nun wird der in der URL stehende Parameter „ch“ von der Methode „parseRequest()“ so umgewandelt wie in Kapitel 4.1.2 beschrieben. Anhand der Konstante kann nun das entsprechende Diagramm erzeugt werden. In diesem Fall „REQ_RClASS“. Dann kann ein neues Objekt der Klasse „RClassChart“ erstellt werden, diesem wird mit Hilfe der Methode „setRelative()“ mitgegeben, ob relative Werte angezeigt werden sollen. Der default ist „false“.
Anschließend wird dem Response mitgeteilt, dass es sich um ein Bild im PNG-Format handelt und das Ganze wird über den „OutPutStream“ an die Klasse „HttpServletResponse“ geschickt, welche das Ganze an den Browser weiterleitet.

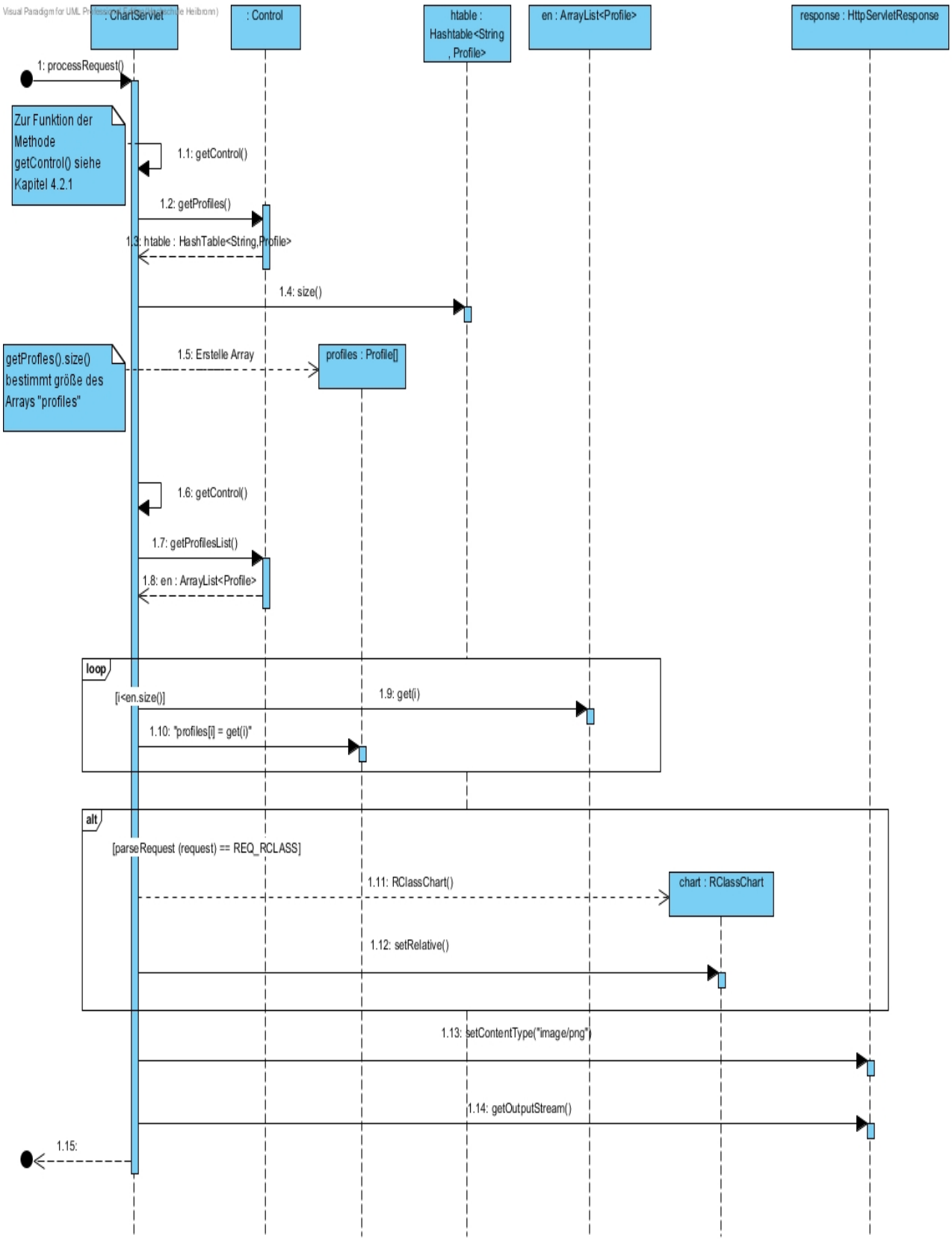

Abbildung 9: Sequenzdiagramm zum Erstellen eines Charts

5. Systemtest

Dieses Kapitel beschreibt den abschließenden Systemtest der Anwendung, um einen reibungslosen Ablauf im Klinikalltag zu gewährleisten.

5.1 Einleitung

5.1.1 Zweck des Tests

Dieser Test dient dazu, die Vollständigkeit der in der Anforderungsanalyse festgelegten Vorgaben zu bestätigen und deren Funktionalität sicher zu stellen.

5.1.2 Testumfang

Der Umfang des Tests beschränkt sich auf das neue Modul „Pankreaskarzinom“.
Dazu gehören:

a) Die 3 verschiedenen Analysemethoden.
b) Das Auswählen der neuen Parameter.
c) Anzeigen der Diagramme im neuen Modul.
d) Vergleich mehrerer Profile.

5.1.3 Referenzierte Unterlagen

Alle Use-Case Diagramme, welche in Kapitel 2, „Anforderungsanalyse“, festgelegt wurden.

5.2 Testumgebung

5.2.1 Überblick

Die Software wird auf einem Rechner getestet auf dem, ein Tomcat 7.0 Server, R 2.15.2 mit den Zusatzpaketen „muhaz“,“splines“ und „RServe“, sowie PostgreSQL 9.2 und ein Browser installiert sind.

5.2.2 Test-Software und -Hardware

Getestet wird das OCDM. Dazu benötigt man einen Browser, einen Tomcat Webserver, die Statistik Software R mit den Erweiterungen „muhaz“, “splines“ und „RServe“ und einen PostgreSQL Datenbank Server.

5.2.3 Testdaten, Testdatenbanken

Es muss eine Datenbank vorhanden sein, die mit den nötigen Tabellen und Funktionen ausgestattet ist. Hierfür stehen entsprechende SQL-Skripte zur Verfügung. Testdaten mit Patientendaten für die Krankheit Pankreaskarzinom müssen von den SLK-Kliniken bereitgestellt werden. Das Skript, um diese Daten in die Datenbank zu laden, wird bereitgestellt. Die Daten für die Sterbetafeln können beim statistischen Bundesamt heruntergeladen werden und als CSV-Dateien in die Datenbank geladen werden.

5.2.4 Personalbedarf

Für die Durchführung des Tests ist eine Person ausreichend.

5.3 Abnahmekriterien

5.3.1 Kriterien für Erfolg und Abbruch

Das Kriterium für den Erfolg ist das vollständige und erfolgreiche durchlaufen aller, weiter unten im Dokument, beschriebenen Tests. Das bedeutet, dass für jeden Test Ist-Ausgabe und Soll-Ausgabe übereinstimmen.
Außerdem muss am Ende des Test eine Testzusammenfassung ausgefüllt werden.

Beim Misserfolg eines Tests, gilt dieser als nicht erfolgreich und die gesamte Testsequenz wird abgebrochen, der verantwortliche Test muss dementsprechend gekennzeichnet werden. Danach kann eine weitere Testsequenz begonnen werden.

5.3.2 Kriterien für Unterbrechungen

Wenn sich die Software während des Verlaufs des Tests ändert, dann muss der Test anschließend wieder von vorne begonnen werden.

5.3.3 Voraussetzungen für Wiederaufnahme

Wenn die Softwarefehler beseitigt sind muss der Test noch einmal von vorne begonnen werden.

5.4 Testabschnitt - Pankreaskarzinom

5.4.1 Einleitung

Dieser Test prüft das Modul „Pankreaskarzinom“ mit den Vorgaben aus der Anforderungsanalyse.

5.4.1.1 Zweck, Referenz zur Spezifikation

Der Test dient dazu, die Lauffähigkeit und Einsatzbereitschaft des Moduls „Pankreaskarzinom“

durch den erfolgreichen Durchlauf der Testsequenzen sicher zu stellen.

5.4.1.2 Getestete Software-Einheiten

Getestet werden die verschiedenen Analysemethoden, die Auswahl der neuen Parameter und das Anzeigen der Diagramme.

5.4.1.3 Vorbereitungsarbeiten für den Testabschnitt

Testdaten müssen, wie in 5.2.3. beschrieben, in die Datenbank geladen werden. Der Datenbank Server muss gestartet werden.
Der R-Server RServe muss gestartet werden. Man muss sich einloggen.

5.4.1.4 Aufräumarbeiten nach dem Testabschnitt

Es sind keine Aufräumarbeiten notwendig, da es keine Schreibvorgänge in die Datenbank gibt.

5.4.2 Testsequenz Subgruppen

Diese Testsequenz umfasst den Test der Analysemethode „Subgruppen“ , dem Vergleich von zwei Profilen und dem Anzeigen aller Diagramme.

Nachfolgend werden die Testschritte der jeweiligen Testfälle beschrieben.

5.4.2.1 Testfall – Neuen Subgruppen-Fall erstellen

Anweisung:

Man klickt oben im Menü auf den Tab „Pankreaskarzinom“.
Man wählt bei „Analysemethode“ den Radiobutton „Subgruppen“ aus.
Man wählt bei „Anzahl der Profile“ den Radiobutton „2“ aus.
Man wählt bei Profil 1, bei „Art der Operation“, „totale Duodenopankreatomie“ aus.
Man wählt bei Profil 2, bei „Art der Operation“, „partielle Duodenopankreatomie“ aus.
Man wählt bei Profil 1, bei „UICC-Klassifikation“, „0“ aus.
Man wählt bei Profil 2, bei „UICC-Klassifikation“, „IA“ aus.
Man klick unten auf der Seite auf den Button „Absenden“.

Soll-Ausgabe:

Es erscheint eine Meldung die sagt, „Bitte Warten“. Anschließend gelangt man auf eine Seite in der man oben die verschiedenen Parameter sieht. Unten ist ein Diagramm zu sehen, sowie Buttons mit denen man andere Diagramme anzeigen kann.
In den allgemeinen Informationen oben steht unter dem Punkt „OP-Art“ bei Profil 1: „totale Duodenopankreatomie“ und bei Profil 2: „partielle Duodenopankreatomie“. Unter dem Punkt „UICC“ steht bei Profil 1: „0“ und bei Profil 2: „IA“.

Ist-Ausgabe und Befund

5.4.2.2 Testfall – Diagramme anzeigen

Anweisungen:

Man klickt auf den Buttons unten alle Diagramme nach einander durch.

Soll-Ausgabe:

Es wird das Diagramm, das man zur Zeit ausgewählt hat, angezeigt.

Ist-Ausgabe und Befund:

5.4.3 Testsequenz Ähnliche Fälle

Diese Testsequenz umfasst den Test der Analysemethode „Ähnliche Fälle“ , dem Vergleich von drei Profilen und dem Anzeigen aller Diagramme.

5.4.3.1 Testfall – Neuen Ähnliche Fälle-Fall erstellen

Anweisung:

Man klickt oben im Menü auf den Tab „Pankreaskarzinom“.
Man wählt bei „Analysemethode“ den Radiobutton „Ähnliche Fälle“ aus.
Man wählt bei „Anzahl der Profile“ den Radiobutton „3“ aus.
Man wählt bei Profil 1, bei „R-Klassifikation“, „R0“ aus.
Man wählt bei Profil 2, bei „R-Klassifikation“, „R1“ aus.
Man wählt bei Profil 3, bei „R-Klassifikation“, „R2“ aus.
Man wählt bei Profil 1, bei „Symptomatik bei Erstdiagnose“ , „ja“ aus.
Man wählt bei Profil 2, bei „Symptomatik bei Erstdiagnose“ , „nein“ aus.
Man wählt bei Profil 3, bei „Symptomatik bei Erstdiagnose“ , „Alle“ aus.
Man klick unten auf der Seite auf den Button „Absenden“.

Soll-Ausgabe:

Es erscheint eine Meldung die sagt, „Bitte Warten“. Anschließend gelangt man auf eine Seite in der man oben die verschiedenen Parameter sieht. Unten ist ein Diagramm zu sehen, sowie Buttons mit denen man andere Diagramme anzeigen kann.
In den allgemeinen Informationen oben steht unter dem Punkt „R-Klassifikation“ bei Profil 1: „R0“ , bei Profil 2: „R1“ und bei Profil 3: „R2“. Unter dem Punkt „Symptomatik bei Erstdiagnose“ steht bei Profil 1: „ja“ , bei Profil 2: „nein“ und bei Profil 3: nichts.

Ist-Ausgabe und Befund:

5.4.3.2 Testfall – Diagramme anzeigen

Anweisungen:

Man klickt auf den Buttons unten alle Diagramme nach einander durch.

Soll-Ausgabe:

Es wird das Diagramm, das man zur Zeit ausgewählt hat, angezeigt.

Ist-Ausgabe und Befund:

5.4.4 Testsequenz Cox-Regression

Diese Testsequenz umfasst den Test der Analysemethode „Cox-Regression“ , dem Vergleich von zwei Parametern und dem Anzeigen aller Diagramme.

5.4.4.1 Testfall – Neuen Cox-Regressions-Fall erstellen

Anweisung:

Man klickt oben im Menü auf den Tab „Pankreaskarzinom“.
Man wählt bei „Analysemethode“ den Radiobutton „Cox-Regression“ aus.
Man wählt bei „Anzahl der Profile“ den Radiobutton „1“ aus.
Man klickt die Checkbox vor „Chemotherapie-Schema“.
Man wählt bei Profil 1, bei „Chemotherapie-Schema“, „Adjuvant“ aus.
Man klickt die Checkbox vor „Substanz der Chemotherapie“.
Man wählt bei Profil 1, bei „Substanz der Chemotherapie“ , „FOLFOX“ aus.
Man klick unten auf der Seite auf den Button „Absenden“.

Soll-Ausgabe:

Es erscheint eine Meldung die sagt, „Bitte Warten“. Anschließend gelangt man auf eine Seite in der man oben die verschiedenen Parameter sieht. Unten ist ein Diagramm zu sehen, sowie Buttons mit denen man andere Diagramme anzeigen kann.
In den allgemeinen Informationen oben steht unter dem Punkt „Chemotherapie-Schema“ bei Profil 1: „Adjuvant“. Unter dem Punkt „Substanz der Chemotherapie“ steht bei Profil 1: „FOLFOX“.

Ist-Ausgabe und Befund:

5.4.4.2 Testfall – Diagramme anzeigen

Anweisungen:

Man klickt auf den Buttons unten alle Diagramme nach einander durch.

Soll-Ausgabe:

Es wird das Diagramm, das man zur Zeit ausgewählt hat, angezeigt.

Ist-Ausgabe und Befund:

6. Abschluss

6.1 Diskussion

Die, im Einleitungskapitel vorgestellte, Zielsetzung konnte vollständig erreicht werden, allerdings konnte der Systemtest nur mit selbst erstellten Testdaten durchgeführt werden. Die Funktionalität mit großen Datenmengen und die Sicherstellung, dass die statistischen Berechnungen richtig sind, kann deshalb nur sehr bedingt bestätigt werden. Das Testen in der Krankenhausumgebung mit realen Patientendaten konnte nicht durchgeführt werden, da das OCDM noch nicht in den SLK-Kliniken installiert werden konnte, bevor die Bearbeitungszeit der Thesis abgelaufen war.

6.2 Ausblick

Das OCDM könnte in Zukunft, genau wie in dieser Arbeit, durch weitere Karzinommodule erweitert werden. Außerdem ist eine Ausweitung der Funktionalität der vorhandenen Module denkbar. Sei es etwa durch mehr Diagramme oder durch die Möglichkeit, die Diagramme genauer zu beeinflussen. Denkbar wäre hier zum Beispiel, dass man festlegen kann, ob – wie bisher - als Anfangsdatum, das Datum der Erstdiagnose oder das Datum des Beginns der Chemotherapie verwendet wird.
Die, im Rahmen dieser Arbeit erstellten Erweiterungen, sowie die gesamte Anwendung, könnten nach einer gewissen Laufzeit evaluiert werden. Man könnte einen Fragebogen an die Benutzer ausgeben, in dem man erfragt, ob sie mit der Benutzung zufrieden sind, was für weitere Funktionalität sie sich wünschen würden oder wie sich ihre Arbeit verbessert hat, falls sie sich denn überhaupt verbessert hat. Die Auswertung dieser Fragebögen könnte dann zu weiteren Änderungen oder Erweiterungen führen, um das System den Bedürfnissen der Nutzer anzupassen. Der Einsatz in weiteren Krankenhäusern ist natürlich auch nicht ausgeschlossen und könnte zu Erweiterungen führen, welche der Natur dieser Arbeit sehr ähnlich sind.

Literaturverzeichnis

01: Peter Fritz and Jürgen Dippon and A. Kleinhans and G. Friedel and D. Zakim and G. Ott and N. Braun and P. Thon and Stefan Winter and W. Brinkmann and D. Alscher, Medizinischer Erkenntnisgewinn aus vernetzten Datenbanken, 2009
02: Klenk S, Dippon J, Fritz P, Interaktive Überlebenszeitanalyse:Datenbankdesign und -implementierung, 2008
03: Sebastian Klenk, Jürgen Dippon, Peter Fritz and Gunther Heidemann, Interactive survival analysis with the OCDM system: from development to application, 2009
04: Sebastian Klenk, OCDM - Interaktive Überlebenszeitanalyse, , http://www.medicaldatamining.de/
05: David Smith, R Tops Data Mining Software Poll, 2012
06: , The R Project for Statistical Computing, , http://www.r-project.org

Abbildungsverzeichnis

Abbildung 1: Grober Aufbau und Kommunikation der Ebenen.....5
Abbildung 2: Paketdiagramm des OCDM.....5
Abbildung 3: Das Frontend des OCDM, mit einem Beispiel Diagramm.....6
Abbildung 4: Konsolen Ausgabe von R, nach Berechnung einer Kaplan-Meier Überlebenskurve.....7
Abbildung 5: Use-Case Diagramm.....8
Abbildung 6: Teil des Klassendiagramms mit Augenmerk auf die Charts 18
Abbildung 7: Die Klassen zur Delegation.....20
Abbildung 8: Sequenzdiagramm zum Holen des „Control“ Objekts 21
Abbildung 9: Sequenzdiagramm zum Erstellen eines Charts.....23

Anhang:

Die folgenden Dateien liegen ausschließlich in der elektronischen Version der Thesis vor.

1. Alle neu erstellten Java-Klassen.
2. Die neu erstellten JSP-Seiten.
3. Die notwendigen SQL-Skripte, um die neuen Tabellen zu erzeugen.
4. Das gesamte Klassendiagramm als Visual Paradigm Projekt, da die Größe kein anderes anderes Format zulässt.
5. Eine CSV-Datei mit einigen Testdaten.