Hochschule für
Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Pflichtenheft
Baumschule - Optimale Entscheidungsbäume

Autoren: Christian Gebauer
Sebastian Große
Benjamin Pfeiffer
Nico Smeenk
Jonathan Wiens

Modul: Software Engineering II (IT2141)

Studienjahrgang: IT2010

Fachrichtung: Informatik

Studienbereich: Technik

Fachbereich: FB 2, Duales Studium Wirtschaft • Technik, HWR Berlin

Stand: 6. Juni 2012

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Vorwort

1.1 Erwartete Leserschaft

Im Allgemeinen können alle Stakeholder des Projektes als Leser erwartet werden, jedoch werden die Entwickler, die Projektleitung, der Auftragnehmer und -geber besonders als Leserschaft hervorgehoben.

1.2 Versionsgeschichte des Dokuments

Dies ist die erste vollständig kompilierte Version des Pflichtenheftes, weitere Versionen sind noch nicht in Planung.

Version	Datum	Autor	Bemerkungen
0.1	30.01.2012	Jonathan Wiens	Struktur angelegt
0.2	14.02.2012	Jonathan Wiens	Inhalte eingefügt: Abschnitte Systemmodelle, Anforderungen an die Entwicklungsumgebung und Ergänzungen
0.4	12.03.2012	Benjamin Pfeiffer	Inhalte eingefügt: Abschnitte Einleitung und Zielbestimmungen
0.5	19.03.2012	Sebastian Große	Inhalte eingefügt: Abschnitte Systemübersicht und -architektur und Systemanforderungen
0.6	02.05.2012	Christian Gebauer	Inhalte eingefügt: Abschnitte Systemeinsatz, Technische Systemumgebung
1.0	14.05.2012	Jonathan Wiens	Inhalte geprüft und kompiliert

2 Einleitung

2.1 Allgemeines

Ein Entscheidungsbaum ist ein speziell aufgebauter Baum aus der Graphentheorie. Im Gegensatz zu einem "normalen"Baum ist ein Entscheidungsbaum geordnet und gerichtet und dient dem Zweck Entscheidungsregeln darzustellen. Ein Entscheidungsbaum kann durch Algorithmen auf Basis von tabellarischen Daten erstellt werden. Aufgrund der Tatsache, dass diese Bäume Entscheidungsprobleme lösen bzw. vereinfachen können, wird diese Methode in vielen Bereichen angewandt. Hauptsächlich benutzt, wird ein Entscheidungsbaum beim Data-Mining, also der umfassenden Theroie der Datenauswertung; aber auch in der Betriebswirtschaftslehre, um effektives Marketing zu betreiben, oder Investment-Geschäfte zu bewerten. Darüber hinaus gibt es viele denkbare Einsatzbereiche, da Entscheidungen fast überall getroffen werden müssen.

Ein Entscheidungsbaum besteht aus Knoten und Blättern. Er muss einen Wurzelknoten besitzen und wird durch Kanten verbunden. Ein Knoten repräsentiert ein(en) Attribut(-namen), Kanten die zulässigen Attributwerte und die Blätter zeigen die Attributwerte (meistens klassifiziert) des Zielattributs an.

In diesem Pflichtenheft wird nun ein Programm beschrieben, dass einen Entscheidungsbaum auf Basis einer Tabelle konstruieren kann. Dieses Programm ist als Auftragsarbeit von Hr. Prof. Dr. Höhne und Fr. Prof. Dr. Monett-Diaz zu sehen, bestimmte Funtktionen und Kriterien sind bereits vorgegeben.

2.2 Abgrenzungen

2.2.1 Ausgangssituation

Im Auftrag von Hr. Prof. Dr. Höhne und Fr. Prof. Dr. Monett-Diaz wird ein Programm, im Rahmen des Moduls SWE-2, entwickelt, welches dazu dienen soll Entscheidungsbäume zu konstruieren und darzustellen. Als Datenbasis dienen Excel- und CSV-Dateien. Mit Hilfe dieser Daten soll dann ein öptimaler Entscheidungsbaumäufgebaut werden, wobei zu beachten ist, dass der Begriff öptimal"nur relativ gesehen werden kann, und abhängig vom gewählten Algorhitmus und der Art des Datenbestandes ist.

2.2.2 Systemname

Aufgrund der Tatsache, dass ein Entscheidungsbaum nach bestimmten Regeln aufgebaut wird und der Anwender einen Lerneffekt aus diesem Baum ziehen kann, in dem Entscheidungen vereinfacht werden, wurde beschlossen das Projekt und das Programm "Baumschule - Optimale Entscheidungsbäumeßu nennen.

2.2.3 Allgemeine Systembeschreibung

Das System bzw. das Programm bietet hauptsächlich drei grobe Funtkionen an:

- Laden/Editieren eines Datenbestands durch eine Datei (Excel bzw. CSV)
- Erstellen eines Entscheidungsbaumes durch einen gegebenen Algorithmus, oder einer Nutzervorgabe
- Präsentation des Ergebnisses

Diese 3 Hauptfunktionen lassen sich näher beschreiben:

- Es gibt eine graphische Benutzeroberfläche mit der der Endandwender alle Funktionen des Programms benutzen kann
- Es gibt eine Funktion zum Laden/Speichern einer Excel- oder CSV-Datei
- Geladene Tabellendaten können editiert werden.
- Ein Baum kann automatisch generiert werden, ein ZielAttribut kann vorher gewählt werden.
- Ein Baum kann teilweise manuell erstellt werden, in dem für jeden Knoten die nächste Attributebene selbst gewählt werden kann.
- Es können jederzeit detaillierte Informationen über einen Baum bzw. dessen Tabellendaten angezeigt werden.

2.2.4 Vorteile und Nutzen

Das System bietet die Bearbeitung eines Datenbestandes an und arbeitet nach dem Laden einer Datei formatunabhängig. Ist ein Datenbestand bearbeitet worden, ist dieser auch wieder exportierbar. Die Oberfläche bietet die Möglichkeit einen, gemäß des Algorithmus, optimalen Baumes darzustellen. Die Stärke liegt hierbei einerseits auf der Präsentation und andererseits bei der leichten Editierbarkeit, denn der Baum kann einfach durch Mausklicks bearbeitet und untersucht werden. Ein manuelles zeichnen und berechnen des Baumes entfällt. Gerade bei großen Daten ist ein Baum nur durch ein Programm wie dieses erstellbar. Arbeitet der Algorithmus fehlerfrei, kann immer wieder und reproduzierbar ein Baum aufgebaut werden.

2.3 Überblick über den Rest des Dokuments

Die nachfolgenden Abschnitte befassen sich detaillierter und gründlicher mit den Funktionen des Programms. Es werden Vereinbarungen festgelegt, welches Verhalten das Programm leisten kann und muss.

3 Fachbegriffe, Akronyme und Abkürzungen

3.1 Glossar

Excel Ein Tabellenkalkulatiosprogramm von Microsoft für Windows und Mac OS

Entropie Die Entropie quantifiziert den erwarteten Wert eines Informationsgehaltes eines Zeichensystems

Tooltip Ein Tooltip ist eine Schnellhilfe, die in Form eines kleines Fensters mit Beschreibungstext, auftaucht

Logdatei Eine Logdatei oder auch Ereignisprotokolldatei registriert bestimmte Prozesse und Aktionen eines Programms automatisch, normalerweise in Textform

Data-Mining Durch oft statistischen Methoden können bei Data-Mining Informationsmuster erkannt und somit Datenmengen analysiert werden

Java Java ist eine objektorientierte Programmiersprache

Plug-in Ein Plug-in ist ein Softwaremodul das oft während der Laufzeit in ein Programm eingefügt werden kann

Eclipse Eclipse ist ein quelloffenes Programmierwerkzeug für Programmiersprachen vieler Arten

Subversive Subversive ist ein Plug-in fur Eclipse zur Versionsverwaltung

XMLBeans XMLBeans ist ein Softwarepaket für Java, dass das Importieren von Informationen aus XML-Dokumenten in Java-Klassen ermöglicht

Apache POI Apache POI ist ein Softwarepaket für Java, dass das Importieren, sowie Exportieren von Dateien im Format von Microsoft Office, bereitstellt

Skype Skype ist eine VoIP-Software von Microsoft mit Instant-Messaging-Funktionen

Java Runtime Environment Mit der Java Runtime Environment, oder Java Laufzeitumgebung, können Java-Anwendung weitgehen unabhängig des darunterliegenden Betriebssystems ausgeführt werden

dom4j dom4j ist eine Schnittstelle zum Verarbeiten von XML Datenstrukturen

CSV Comma Seperated Value

XML Extended Markup Language

PNG Portable Network Graphic

JPG Joint Photographic Experts Group

SVG Scalable Vector Graphic

XLS Excel Spreadsheet

XLSX Excel Spreadsheet 2007

SWT Standard Widget Toolkit

4 Zielbestimmungen

Die folgenden Ziele sind in 3 Kategorien (Muss-, Soll- , Kannkriterien) unterteilt. Durch diese Unterteilung wird eine Prioritätshierarchie geschaffen, wobei Musskriterien die höchste Priorität erhalten und entsprechend vorrangig abgearbeitet werden.

4.1 Musskriterien

Die Muss-Kritierien ergeben sich größtenteils aus der Aufgabenbeschreibung von Hr. Höhne und Fr. Monett-Diaz und sollen hier abgekürzt wiedergegeben werden:

- Das Programm muss in der Lage sein mit einem Datenbestand von bis zu 10 Attributen und 100 Objekten arbeiten zu können.
- Das Programm muss mit Tabellen-Dateien umgehen können. (Laden/Speichern von Excel- und CSV-Dateien)
- Das Programm muss ein Bearbeiten der Tabellendaten zulassen.
- Es muss eine graphische Benutzeroberfläche geben.
- Das Zielattribut muss farblich hervorgehoben werden.
- Das Programm muss eine Möglichkeit bieten den Baum, durch Auswahl des nächsten Attributs, teilweise manuell aufzubauen.
- Wenn der Baum interaktiv zusammengestellt wird, darf(muss) nur der oberste Knoten präsentiert werden.
- Bei der Präsentation des interaktiven Baums kann dieser durch ein Mausklick üntersucht"werden. Es wird bei einem Knoten eine Tabelle angezigt, die die Objekte anzeigt, die durch den Knoten repräsentiert werden.
- Es kann in der Tabelle ein Attribut untersucht werden, woraufhin die gewichtete Entropie, die sichbei Aufteilung nach diesem Attribut ergeben würde, angezeigt wird.
- Das Programm muss anhand eines vorgegebenen Algorithmus (Konzept der Entropie) einen Entscheidungsbaum erstellen können.
- Bei der automatischen Erstellung wird ein Knoten der nur noch Objekte einer Klasse enthält nicht weiter aufgeteilt, selbiges gilt wenn der Knoten eine (einstellbare) Anzahl von Elementen enthält.
- Bei der Präsentation des autmatischen Baums kann dieser durch ein Mausklick üntersucht"werden. Es wird bei einem Knoten eine Tabelle angezigt, die die Objekte anzeigt, die durch den Knoten repräsentiert werden.
- Bei der Präsentation des interaktiven und autmatischen Baums müssen die Kanten mit dem entsprechenden Attributewert angezeigt werden.
- Bei der Präsentation des interaktiven und automatischen Baums muss die Zahl der enthaltenen Objekte, die Zahl jeder Klasse und die Entropie in jedem Knoten angezeigt werden.

- Der Entscheidungsbaum muss, egal ob manuell oder autmoatisch erstellt, bei Änderungen des Datenbestandes aktualisiert werden.
- Das Programm muss lauffähig sein und ausführbar vorliegen.

4.2 Sollkriterien

- Das Programm soll an den wichtigen Stellen durchdachte Hilfebeschreibungen(ToolTips) anzeigen können.
- Das Programm soll für den Benutzer ein vollständiges Handbuch bereitstellen, dass jederzeit über das Menü aufgerufen werden kann.
- Das Programm soll bei bestimmten Fehlern oder unerlaubten Nutzeraktionen eine graphische Fehlermeldung anzeigen.
- Das Programm soll zum Nachverfolgen von Programm-Algorithmen und -Abläufen eine Log-Datei erstellen.
- Die Oberfläche soll für den Nutzer farblich und organisatorisch übersichtlich und leicht zu bedienen sein.
- Das Erstellen eines Baumes soll nicht länger als 5 Sekunden dauern. (Vorrausgesetzt das Datenlimit von 10*100 Attributwerten wird eingehalten)
- Die Oberfläche soll neben der einfachen Ansicht einen Präsentationsmodus bieten, in dem der Baum als Vollbild dargestellt wird.
- Für die Anzeige des Baumes soll eine Legende bereitgestellt werden, die den Aufbau eines dargestellten Baumes erklärt.

4.3 Kannkriterien

- Das Programm kann die Möglichkeit bieten, über eine Konfigurationsschnittstelle verschiedene Programmparameter anzupassen.
- Für den Benutzer kann es eine grafische Möglichkeit geben die Konfigurationsschnittstelle zu nutzen.
- Der Nutzer kann die Möglichkeit erhalten den dynamisch erstellten Baum als XML-Datei (*.xml) zu speichern.
- Der Nutzer kann die Möglichkeit erhalten die abgespeicherte xml-Datei wieder zu laden; die Oberfläche stellt dann automatisch die Tabelle und die Bäume wieder her.
- Das Programm kann die Möglichkeit bieten den dynamischen und automatischen Baum als Bild-Datei (*.png ; *.jpg) abzuspeichern.
- Das Programm kann die Möglichkeit bieten den dynamischen und automatischen Baum als Vektorgrafik (*.svg) abzuspeichern.
- Die Zeichenoberfläche, in dem der Baum (dynamisch oder manuell) präsentiert wird, kann durch einen Schieberegler heran- bzw. herausgezoomt werden.

4.4 Abgrenzungskriterien

- Die Software setzt eine installierte Version von Java voraus.
- Es gibt für jedes Betriebssystem (Windows, Linux, Mac) und dessen Ausprägung (x64, x32) jeweils eine Version.
- Die Software kann nur mit Datenformaten umgehen, die Excel-Konform sind- (bis 2003: .xls, ab 2007: .xlsx) oder dem CSV-Format entsprechen (Trennzeichen: ","(Ausgesprochen: Komma)
- Die Software benutzt einen fest vorgegebenen Algorithmus (Konzept der Entropie).
- Die Tabelle kann maximal 10 Attribute und jeweils 100 Attributwerte enthalten und darf innerhalb der Tabellengrenzen keine leeren Zellenwerte (leere Strings) enthalten.

5 Systemeinsatz

5.1 Anwendungsbereiche

Das Programm soll eingesetzt werden, um einen optimalen Entscheidungsbaum zu finden. Optimale Entscheidungsbäume werden benötigt, um aus einem gegebenen Datenbestand Voraussagen zum zukünftigen Verhalten treffen zu können. Speziell sollen diese optimalen Entscheidungsbäume mit der Software präsentiert werden können. In Vorlesungen an der Hochschule soll das Programm u.A. von Dozenten zur Visualisierung der Konzepte beim Data-Mining eingesetzt werden.

5.2 Zielgruppen

Als Zielgruppe des Programmes sind in erster Linie die Auftraggeber Hr. Prof. Dr. Höhne und Fr. Prof. Dr. Monett-Diaz zu sehen. Darauf folgend voraussichtlich aber auch die Dozenten und Studenten der HWR Berlin, speziell vorrangig aus der Fachrichtung Informatik.

5.3 Betriebsbedingungen

Folgende Bedingungen sind für den Betrieb des Programmes relevant:

- Der Nutzer ist an einem aktuellen PC mit Windows-Betriebssystem (alternativ auch Linux oder Mac) angemeldet.
- Auf dem System ist eine Java-Umgebung vorhanden.
- Optimalerweise hat der PC ein für Präsentationen geeignetes Ausgabegerät (z.B. Beamer, genügend großer Monitor).

6 Technische Systemumgebung

6.1 Software

- Das Programm soll auf allen gängigen Windows-Betriebssystemen lauffähig sein.
- Es soll eine Version für 32- und eine für 64 Bit Systeme geben.
- Es sollen weitere Betriebssysteme (Linux, Mac) unterstützt werden (optional).
- Eine Java-Umgebung muss auf den Zielsystemen vorhanden sein.
- Das Programm soll ohne Installation lauffähig sein.

6.2 Hardware

- Es wird ein aktuell üblicher Standard-PC vorausgesetzt (auf dem Windows mit Standardsoftware lauffähig ist).
- Für ein flüssiges Abarbeiten und Darstellen der relevanten Inhalte ist in erster Linie ein ausreichend dimensionierter Arbeitsspeicher relevant , hier sollten mindestens 2 GB verfügbar sein.
- Es wird ein präsentationsfähiges Ausgabegerät (großer Monitor, Beamer, etc.) benötigt, da das Programm u.A. für Präsentationen eingesetzt werden soll.

6.3 Orgware

- Es wird keine explizite Orgware benötigt. Der Anwender kann alle relevanten Einstellungen selbst direkt in der Software bzw. in der Konfigurations-Datei vornehmen.

7 Systemübersicht und -architektur

7.1 Übersicht über das System

Das Programm besteht hauptsächlich aus einem Controller für die Funktionalität, der View für die Interaktion mit dem Nutzer und dem Model zum Verwalten der Daten. Desweiteren wird ein Observern Pattern verwendet.

Das folgende Diagramm stellt die Hauptanwendungen des Benutzers dar:

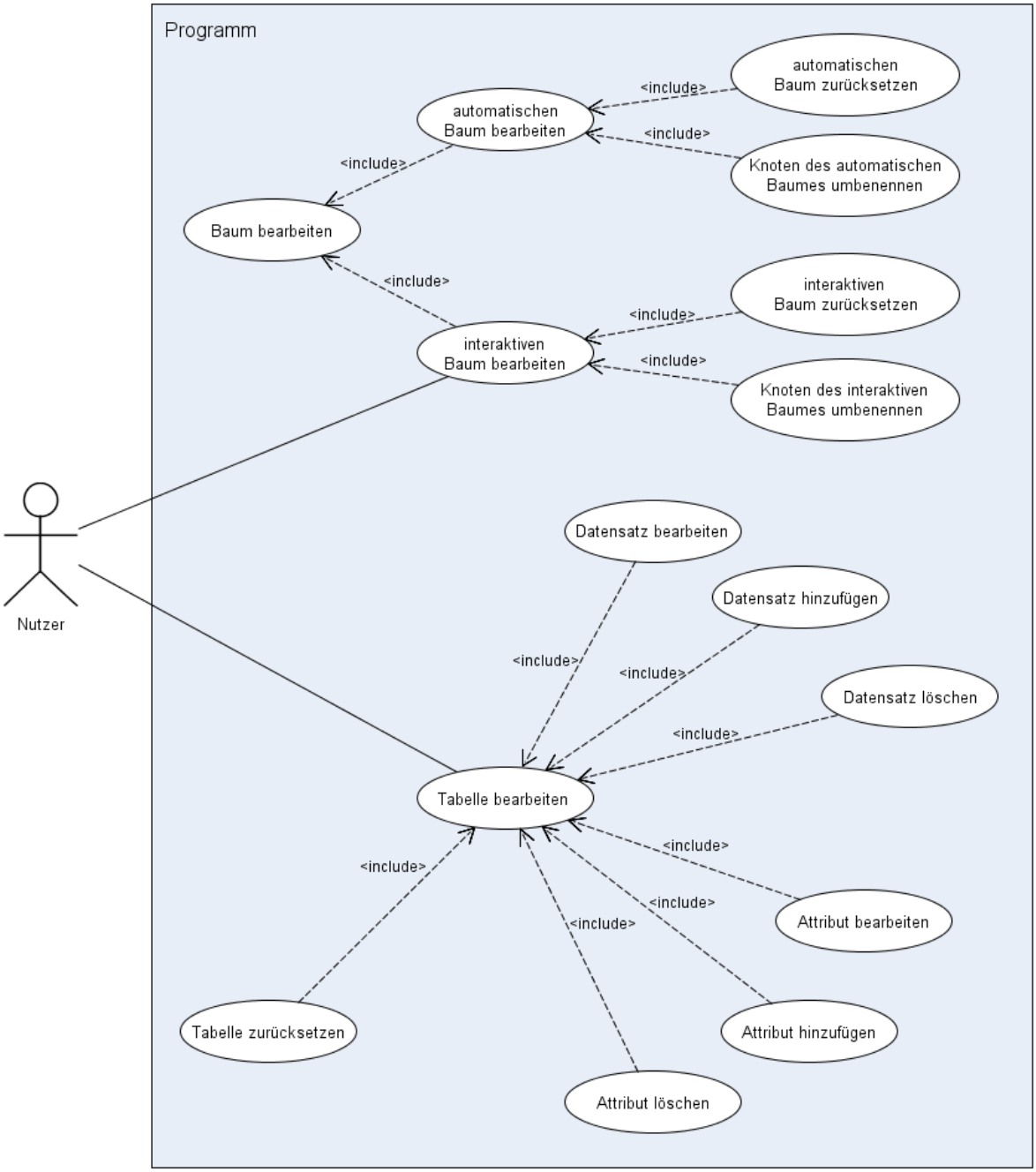

Abbildung 1: Programmnutzung des Nutzers in Form eines UseCase-Diagramms

Um die zwei Hauptaktivitäten des Benutzers klarer darzustellen, dienen die zwei folgenden Aktivitätsdiagramme nach Anforderungsschablone nach Rupp:

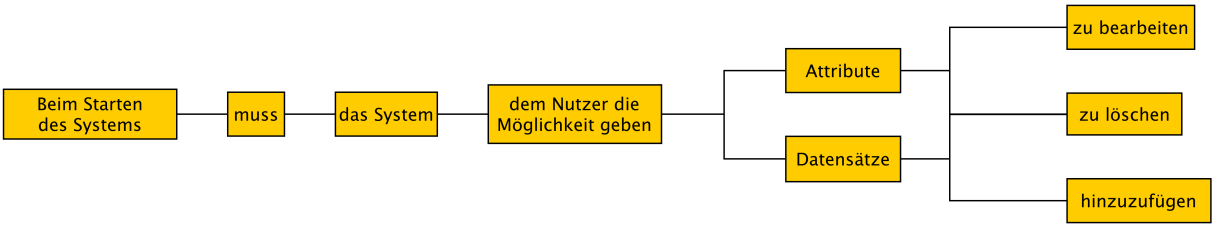

Abbildung 2: Veranschaulichung wann der Nutzer die Tabelle bearbeiten darf in Form eines Aktivitätsdiagramms nach Anforderungschablone von Rupp

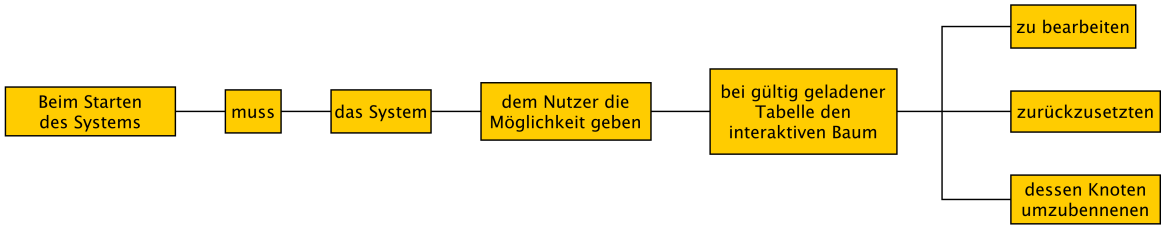

Abbildung 3: Veranschaulichung wann der Nutzer die den interaktiven Baum bearbeiten darf in Form eines Aktivitätsdiagramms nach Anforderungschablone von Rupp

7.2 Systemarchitektur

Das folgende Klassendiagramm zeigt, in vereinfachter Weise, die komplette Systemarchitektur. Es wurde das Observer-Pattern benutzt. Verschiedene statische und view-Klassen wurden weggelassen.

Die wichtigsten Elemente der Architektur sind:

- die View,
- das Model,
- und der Controller.

7.3 Schnittstellen des Systems

7.3.1 View

Die Schnittstelle zwischen dem Programm und dem Nutzer wird von der View gestellt Die View/GUI ist zur grfischen Darstellung und zur dierekten interaktion mit dem Nutzer verantwortlich. Neben der

- Hauptklasse "View"

gehören noch die Klasse

- "TableView"(die Tabellenansicht),
- "TreeView“ (die Baumansichten),
  - "TreeZoomView"(der Zoom eines Bäume),
    - "TreePainter"(die gezeichnete darstellung eines Baumes)

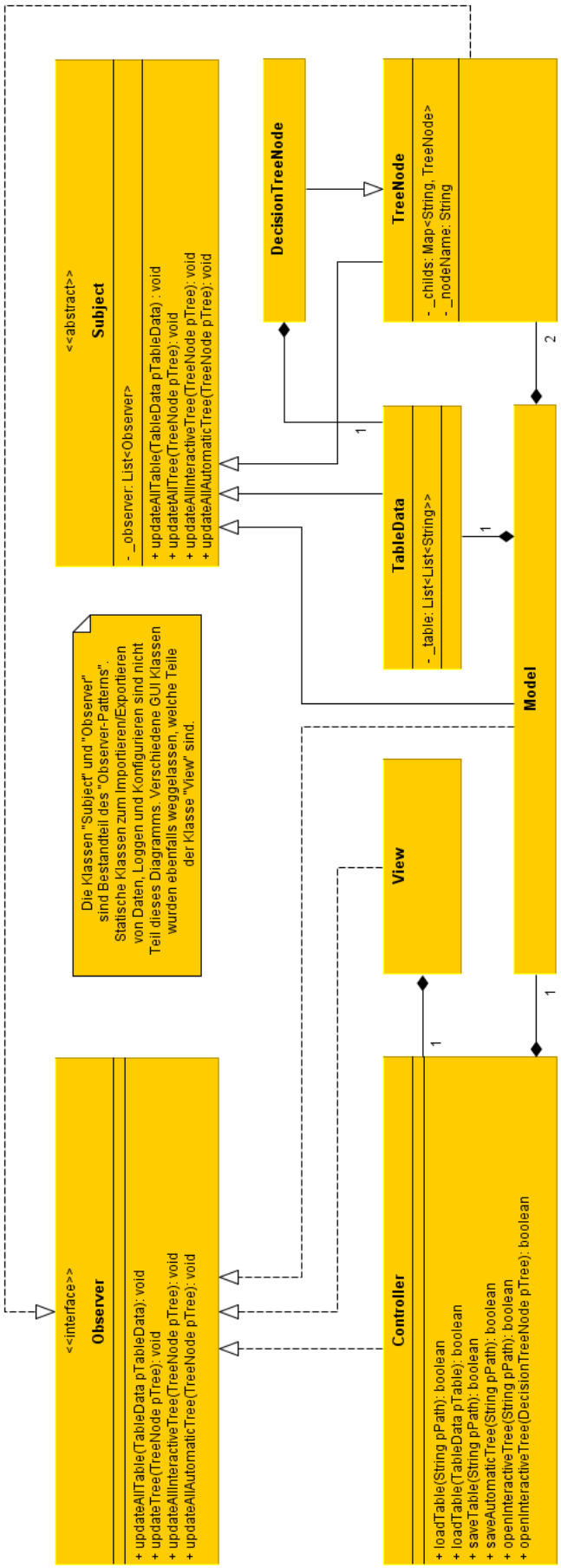

Abbildung 4: Ein Klassendiagramm mit den wichtigsten Klassen zur Architektur

  - "TreeDetails"(die Informationen eines Ausgewählten Knotens, die neben dem Baum dargestellt werden)

- "FullScreenTreeView"(die Baumansicht für den Presentationsmodus)

dazu. Die Klassen View, TableView und TreeView beobachten jeweils als Observer das Model.

7.3.2 Model

Das Model besitzt die Tabelle, den interaktiven Baum und den automatischen Baum. Es beobachtet als Observer die Tabelle und die Bäume und wird als Subject vom Controller und von mehreren Klassen der View beobachtet. Es dient somit als Schnittstelle zwischen dem Controller und den Daten (Tabelle und Bäume).

7.3.3 Controller

Der Controller ist eine Klasse, die die Schnittstelle zwischen GUI und Funktionalität herstellt, damit diese beiden Bereiche nicht vermischt werden. Dabei stellt der Controller selbst auch einen Teil der Funktionalität dar. Er beobachtet als Observer das Model. Die View spricht den Controller an, wenn die Tabelle bearbeitet, eine Tabelle oder ein Baum geladen oder eine Tabelle oder ein Baum gesteichert werden soll. Der Controller greift dann auf die zuständigen Klassen zu. Somit muss die View immer nur auf dne Controller zugreifen und wird nicht weiter mit der Funktionalität des Programmes verstrickt.

8 Systemanforderungen

Das Programm soll den Argorithmus, einen optimalen Entscheidungsbaum zu finden, demonstrieren.

8.1 Funktionale Anforderungen

- Das Programm soll unter allen gängigen Windows-Betriebssystemen ohne Installation lauffähig sein (Java darf vorrausgesetzt werden, sonst nichts spezielles). [Bearbeiten]

8.1.1 Tabellenansicht

- Einlesen / Speichern einer Tabelle im .csv-Format (wobei das Trennzeichen ein Komma ist).
- Alle Attribute sind Zeichenketten.
- Eingeben / Editieren einer Tabelle muss möglich sein.
- Import und Export von Excel-Dateien.

8.1.2 Baum automatisch

- Der optimale Entscheidungsbaum wird mit dem Algorithmus, der sukzessiv für jeden Knoten jeweils das Attribut mit der minimalen gewichteten Entropie auswählt, erzeugt und dargestellt.
- Wenn ein Knoten nur noch Objekte einer Klasse enthält, wird dieser nicht weiter aufgeteilt.
- Wenn ein Knoten nur noch eine bestimmte Anzahl (einstellbar - Voreinstellung 1) von Elementen enthält, wird nicht weiter aufgeteilt.

8.2 Benutzeroberfläche

Weil das Programm auch für Präsentationszwecke eingesetzt werden soll, ist insbesondere auf eine entsprechende Gestaltung der Oberfläche (aussagekräftige Verwendung von Farben, ausreichend großer Schriftgrad, etc.) zu achten. Im Programm kann zwischen drei verschiedenen Ansichten hin- und hergeschalten werden, von denen stets genau eine sichtbar ist:

- Tabellenansicht
- Baum interaktiv
- Baum automatisch

Start ist stets in der Tabellenansicht, erst wenn eine Tabelle geladen oder eingegeben ist, kann in eine Baumansicht gewechselt werden. Ansonsten kann stets zwischen den drei Ansichten umgeschalten werden.

Falls keine korrekte Tabelle geladen ist, wird der Nutzer informiert und die Baumansichten sind automatisch deaktiviert und werden erst nach bearbeiten oder Laden einer richtigen Tabelle aktiviert. Dazu das folgende Zustandsdiagramm:

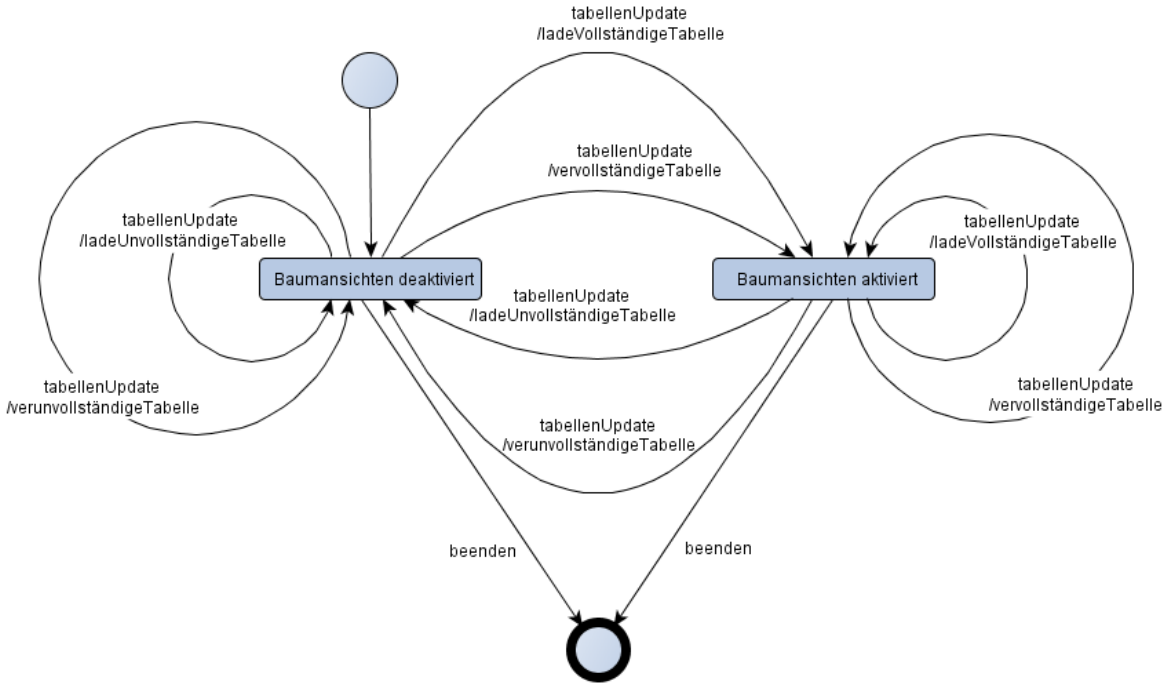

Abbildung 5: Ein Zustandsdiagramm zur Veranschaulichung wann welche Benutzeransicht de- bzw. aktiviert ist.

8.2.1 Tabellenansicht

- Eine Tabelle besteht aus maximal etwa 10 Attributen und 100 Objekten. Die Spalten repräsentieren dabei die Attribute und die Zeilen die Objekte / Datensätze.
- (Farbliche) Markierung des Zielattributs.

8.2.2 Baum interaktiv

- Zu Beginn wird nur der oberste Knoten (also der, der allen Objekten in der Tabelle entspricht) dargestellt.
- Das gilt ebenso, wenn in der Tabellenansicht ein Attribut hinzugefügt oder gelöscht oder das Zielattribut geändert wird.
- In jedem Knoten ist die Zahl der enthaltenden Objekte, die Zahl der Objekte jeder Klasse und die Entropie angegeben.
- Die Kanten sind mit dem entsprechenden Attributwert markiert.
- Bei Identifikation eines Knotens des Baumes wird eine Tabelle mit den Objekten, die von diesem Knoten repräsentiert werden, dargestellt.
- Wird in dieser Tabelle ein Attribut identifiziert, wird die gewichtete Entropie, die sich bei Aufteilung nach diesem Attribut ergeben würde, ausgegeben.
- In dieser Tabelle kann ein Attribut selektiert werden, so dass entsprechend den Werten dieses Attributes Unterknoten entstehen. Die Tabelle verschwindet. Hatte der Knoten schon Unterknoten, verschwinden diese natürlich ebenfalls.

8.2.3 Baum automatisch

- In jedem Knoten ist die Zahl der enthaltenden Objekte, die Zahl der Objekte jeder Klasse und die Entropie automatisch dargestellt.
- Die Kanten sind mit dem entsprechenden Attributwert markiert.
- Bei Identifikation eines Knotens des Baumes wird eine Tabelle mit den Objekten, die von diesem Knoten repräsentiert werden, dargestellt.

8.3 Qualitätsanforderungen

- Funktionalität:
  - Alle Anforderungen müssen erfüllt sein, so wie sie beschrieben sind. Kommunikation mit dem Kunden sollen verschiedene Interpretationen der Anforderungen elimieren
- Zuverlässigkeit
  - Das Programm darf keine Abstürze haben und stabil laufen. Dabei ist darauf zu achten, dass die File I/O im Rahmen des Im- und Exports daraufhin geprüft werden.
- Benutzbarkeit
  - Die Funktionen des Programms sollen so weit wie möglich intuitiv sein. Desweiteren soll ein Handbuch existieren, falls der User Unterstützung braucht. Außerdem sollen Schnellhilfen wie Tooltips und Legenden zur Verfügung stehen
- Effizienz
  - Das Programm soll eine Reaktionsgeschwindigkeit von unter 2 Sekunden haben. Falls große Bilder exportiert werden, soll dies je nach Auflösung bis zu 7 Sekunden dauern.
- Änderbarkeit
  - Das Programm soll modular und objektorientiert aufgebaut sein, sodass mögliche Versionserweiterungen angeschlossen werden können. Außerdem soll es eine Entwicklerdokumentation geben, die es Neuentwicklern die möglichkeit gibt die Strukturen des Programms besser kennen zu lernen
- Übertragbarkeit
  - Da die Software in Java programmiert wird, soll es in möglichst viele Umgebungen übertragen werden können. Alle Betriebssysteme sind dabei ein Muss.

8.4 Andere Nicht-funktionale Anforderungen

Da es sich um ein reines Studienprojekt handelt und keine Benutzerdaten aufgenommen werden müssen, brauchen keine besonderen Gesetze, Normen, Testate, etc. eingehalten werden. Die einzige nicht-funktionale Anforderung ist die bereits erwähnte Plattformunabhängigkeit.

9 Systemmodelle

So wie es für den Benutzer insgesamt drei Hauptansichten gibt, so gibt es auch intern drei verschiedene Ebenen, die das Model, bzw. die Datenstrukturen, über mögliche Änderungen informiert:

- Falls der Benutzer etwas in der Tabelle ändert wird das Model, bzw. werden die Datenstrukturen informiert und gegebenfalls bearbeitet.
- Somit wird eine Kette von Aktivitäten ausgeführt (siehe das Diagramm unten)
- Nachdem die Datenstruktur geändert wurde, wird die grafische Oberfläche, oder auch die View, informiert, damit sich diese entsprechenderweise aktualisiert

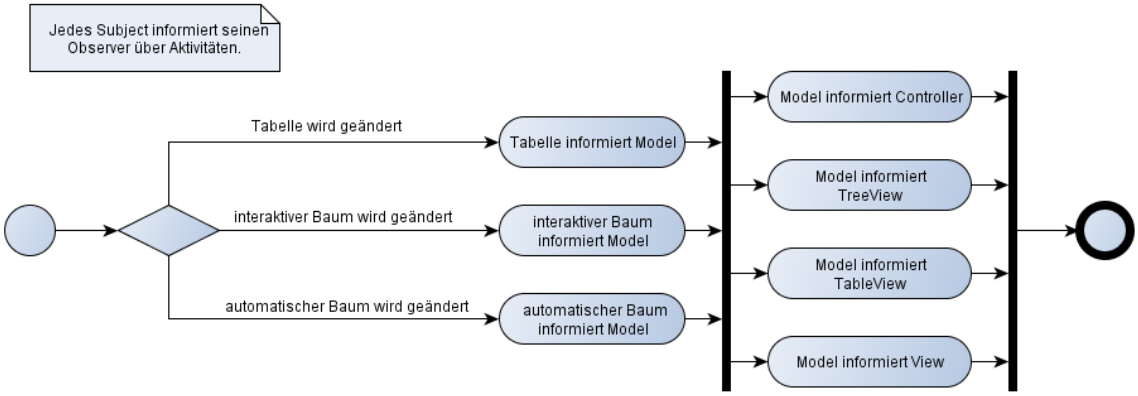

Abbildung 6: Ein Aktivitätsdiagramm zur Veranschaulichung des Observerprinzips

9.1 Datenmodelle

Die Daten die in dem Projekt verarbeitet werden sollen, liegen zunächst in Form einer Tabelle vor. Diese Tabelle muss in Java abgebildet werden. Diese Datenorganisation bildet dann auch die Schnittstelle zu unseren Export- und Import-Klassen. In einem weiteren Schritt müssen die Daten noch in eine Baumstruktur gewandelt werden. Jeder Zeile ist dabei ein Listenelement, welches wiederum eine Liste ist, das allen zu dieser Zeile gehören Spaltenattribute enthält. Es gibt also eine Liste in einer Liste und kann daher ähnlich angesprochen werden wie ein Koordinatensystem.

Für Bäume gibt es eine Klasse, die einen Knoten repräsentiert. Ein solcher Knoten hat als Attribute einen Namen, ein Daten-Objekt und eine Liste von weiteren Knoten, die dann seine Kind-Knoten darstellen. Um einfacher mit einem Baum arbeiten zu können, ist in jedem Knoten zusätzlich auch der Eltern-Knoten hinterlegt. Um den Baum anzusprechen oder zu kopieren, wird der Root-Knoten angesprochen.

Um Entscheidungsbäume darzustellen wird eine Klasse für dessen Knoten implementiert, welche von der Klasse den Baumknoten erbt. Diese speichert zusätzlich noch die zugehörige Tabelle, den Attributnamen des Knoten und das Ziel-Attribut. Weiter bietet es Methoden zur Berechnung der Entropie, zur Berechnung der gewichteten Entropie und zur Erstellung eines optimalen Entscheidungsbaumes, ausgehend von sich selbst.

9.2 Andere Modelle

Das folgende Diagramm zeigt auf, nach welchem Modell der optimierte Baum erstellt wird. Dabei wird eine Reihe von Bedingungen geprüft und Rechnungen durchgeführt:

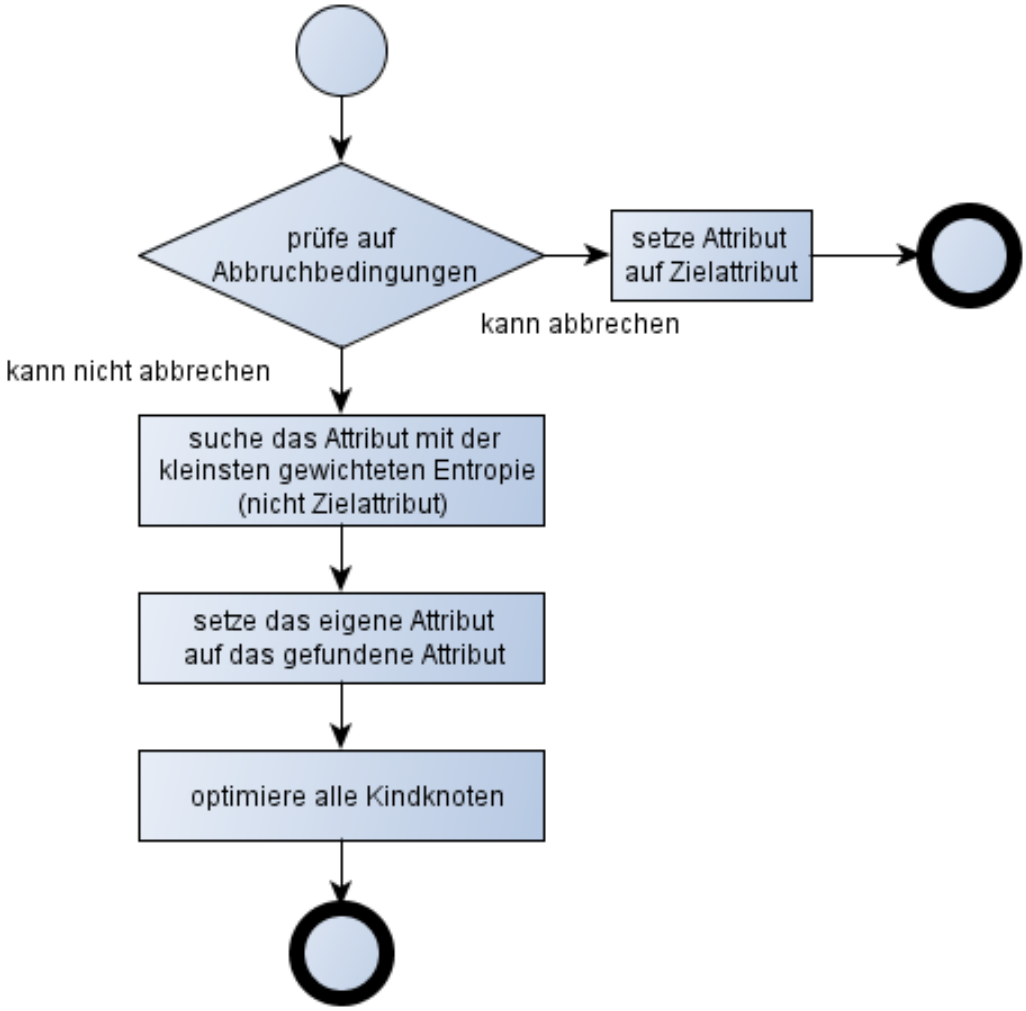

Abbildung 7: Ein Flussdiagramm zur Veranschaulichung zur Erstellung eines optimierten automatischen Baumes

10 Anforderungen an die Entwicklungsumgebung

Die hier genannten Aspekte werden für die qualitative Entwicklung der Software benötigt.

10.1 Software

Das Betriebssystem spielt keine Rolle, da Eclipse und Java Plattformunabhängig sind. Die Prozessorarchitektur (32bit oder 64bit spielt demnach auch keine Rolle)

- Die Software Ëclipseßur Programmierung mit Java wird benötigt. Hinzu kommt das Eclipse PlugIn ßubversiveßur Handhabung der Versionsverwaltung
- Eine aktuelle Java-Developer-Version (jdk) wird zur Kompilierung benötigt. Außerdem werden folgende externe Bibliotheken benötigt:
  - "dom4j", Äpache POIßur Konvertierung und Erstellung von Excel-Spreadsheet Tabellen
  - SSWT", das genutzte Framework zur Erstellung der grafischen Benutzeroberfläche
  - "XMLBeansßur Erstellung und Lesen von XML-Dateien

10.2 Hardware

Für die verwendete Hardware sollten mindestens 2 GB Arbeitsspeicher verfügbar sein, um ein flüssiges Abarbeiten und Darstellen der relevanten Inhalte zu gewährleisten.

10.3 Orgware

Vorrangig geschieht die Kommunikation im Team über E-Mail Nachrichten und Kurzmitteilungen. Es sollte die VoIP-Software Skype installiert sein und ein Account bereitstehen, um einfache und schelle Kommunikation zu ermöglichen.

11 Ergänzungen

Das Programm sollte auf allen gängigen Betriebssystemen, Mac OS X, Windows 7 und Linux ausführbar sein.

Zwar ist die Java Runtime Environment, mit der das Programm plattformunabhängig ausführbar ist, für jedes der obigen Betriebssysteme verfügbar, jedoch muss trotzdem auf Betriebsbedingte Eigenheiten beim Programmieren geachtet werden. Zum Beispiel sind einige Elemente der Fenster bei Mac OS X anders als bei Windows. Somit muss beim Programmieren für jedes Betriebssystem eigene Tests durchgeführt werden, besonders bei dem Aspekt der graphischen Oberfläche.

Um erfolgreich die Software auf den verschiedenen Betriebssystemen zu installieren, muss für jedes Betriebssystem ein eigener Erstellungsprozess mit der Betriebssystemspezifischen SWT-Bibliothek ausgeführt werden. Mac OS X, zum Beispiel, brauch eine andere SWT-Bibliothek als Linux, da Mac OS X eine andere Umgebung für die graphische Oberfläche hat.