Studienarbeit

Entwicklung eines Systems zur Clusteranalyse von Benutzerprofilen

vorgelegt von
Christian Butrynowski
am 8. Februar 2005

Studiengang Softwaretechnik
Betreuer: Prof. Dr. Kai von Luck
Inhaltsverzeichnis

1 Einleitung ........................................ 4
  1.1 Motivation ................................... 4
  1.2 Gliederung ................................... 4

2 Szenario ......................................... 6
  2.1 Kalle im Urlaubsc lub .......................... 6

3 Analyse .......................................... 7
  3.1 Rollen ......................................... 7
       Administrator .................................. 7
       Feriengast .................................... 7
  3.2 Use Cases ...................................... 8
       Administrator .................................. 8
       Feriengast .................................... 10
  3.3 Nähe und Distanz ............................... 10
       Euklidische Distanz ............................ 11
       Gewichtete Euklidische Distanz ............... 12
       Fuzzy-Logik .................................... 14
  3.4 Optimierungsproblem ......................... 16
       Reihenfolgeproblem ............................ 16
       Clusteranalyse ................................ 18
       Fazit .......................................... 19

4 Design und Implementierung .................. 20
  4.1 Systemarchitektur ............................. 20
       Java ........................................... 20
       Model-View-Controller ......................... 21
  4.2 Komponenten .................................. 22
  4.3 Klassenmodelle ............................... 23
       Web-Komponente ................................ 23
       Komponente Profillverwaltung ................. 24
Inhaltsverzeichnis

Clusterung-Komponente .................................................. 24

5 Fallstudie 27
  5.1 Startseite ............................................................... 27
  5.2 Administrationsmenü ................................................... 27
  5.3 Profilvorlage ............................................................ 27
  5.4 Profil ausfüllen ......................................................... 28
  5.5 Zufallsprofile erzeugen und alle Profile anzeigen .............. 29
  5.6 Profile in Cluster einteilen ........................................... 31

6 Ausblick 34
  6.1 PDA ................................................................. 34
  6.2 Sicherheit .............................................................. 34
  6.3 Persistenz ............................................................. 35

Literaturverzeichnis 36
1 Einleitung

1.1 Motivation

Computer spielen seit Jahren eine zunehmend größere Rolle in unserem Alltag. Mittlerweile besitzen viele Menschen nicht bloß einen PC, sondern viele weitere Computer - oftmals ohne sich dessen bewußt zu sein. Dazu gehören Mobiltelefone, die mit jeder Generation weitere Funktionen bieten welche weit über das Telefonieren hinaus gehen. Dazu zählen digitale Videorecorder, Navigationssysteme in Fahrzeugen, MP3-Player oder auch PDAs\textsuperscript{1}.


In dieser Arbeit soll eine mögliche Anwendung für den Einsatz eines PDAs in einem Urlaubssclub entwickelt werden. Die Anwendung soll dazu dienen Menschen mit ähnlichen Interessen leichter zusammenzubringen und ihren Urlaub damit zu verschönern.

1.2 Gliederung

Im Kapitel 2 soll ein mögliches Szenario für die Anwendung gezeichnet werden, um dem Leser eine bildhafte Vorstellung zu ermöglichen. In Abschnitt 3 wird die Aufgabenstellung analysiert. Es werden die Rollen und Anwendungsfälle vorgestellt, und es werden algorithmische Lösungsansätze gezeigt. Das folgende Kapitel 4 befasst sich mit dem Design und der Implementierung der Anwendung. Dabei wird zunächst die grobe Systemarchitektur vorgestellt um schließlich in die Tiefe zu gehen und am Ende das Klassenmodell vorzustellen. Die Gestaltung und die Benutzung der Anwendung ist Thema von Abschnitt 5, in dem die Anwendung anhand einer konkreten Fallstudie vorgestellt wird. Zu guter Letzt gibt das

\textsuperscript{1}Personal Digital Assistant
Kapitel 6 einen **Ausblick** auf unbearbeitete Aspekte von get2gether, die in einer endgültigen Version Beachtung finden müßten.
2 Szenario

Das folgende Szenario soll veranschaulichen, was get2gether leisten soll. Auch wenn im Rahmen dieser Arbeit nicht alle Aspekte vollständig umgesetzt werden, soll das Szenario als Vision einer möglichen Fortentwicklung des Projektes dienen.

2.1 Kalle im Urlaubsclub

Single Kalle möchte in Urlaub fahren und dabei gerne Leute kennenlernen mit denen er seine liebsten Freizeitaktivitäten teilen kann. Da entdeckt er in einer Broschüre des Urlaubclubs get2gether folgenden Satz: “Urlaub unter "seinesgleichen": unkompliziert, ausgelassen und relaxt in einer Gruppe von Gleichgesinnten...”. Kurzentschlossen nimmt er das Angebot wahr und erreicht wenig später auch schon den modernen Urlaubscclub get2gether. Als Kalle im Hotel ankommt, erhält er an der Rezeption neben seinem Zimmerschlüssel auch einen PDA. Die Dame bittet ihn dabei freundlich, zwecks optimaler Urlaubsgestaltung, doch mal einen Blick auf die Software get2gether zu riskieren.

Neugierig und hoffnungsträger lässt sich Kalle nicht zweimal bitten. Das Programm fragt ihn nach einigen persönlichen Daten wie Geschlecht, Alter und ob er Single sei. Anschließend wird er aufgefordert einige potentielle Interessengebiete auf einer Skala von 1 bis 10 einzuziehen. Die Interessenliebhaber umfassen dabei u.a. allerlei Sport- und Freizeitaktivitäten, die im Club angeboten werden (Golfen, Reiten, Segeln, Surfen, Tanzen...). Nachdem er alle Eingaben getätigt hat, erhält Kalle die Rückmeldung, dass seine Daten übertragen und gespeichert werden. Kurz darauf macht ihm get2gether den Vorschlag sich beim Abendessen doch mal an Tisch 7 zu setzen. Dort säßen weitere Personen die ähnlichen Interessen wie er nachgingen. Nun möchte Kalle die weiteren Funktionalitäten ausprobieren und gibt an, dass er gerne morgen um 14.00 eine Partie Tennis spielen möchte, und ob ihm der Assistent nicht einen Spielpartner aussuchen könnte. Kurze Zeit später erscheint auf dem PDA des tennisbegeisterten Klaus die Frage, ob er nicht Lust hätte morgen um 14.00 eine Partie Tennis gegen den ebenso tennisbegeisterten Kalle zu spielen. Klaus ist davon sehr angetan, antwortet natürlich mit Ja, woraufhin Kalle eine Bestätigung erhält.
3 Analyse

In diesem Abschnitt sollen zum Einen die Rollen beschrieben werden, die die Benutzer des Systems annehmen können. Zum Anderen wird anhand von Anwendungsfällen ("Use Cases") aufgezeigt, wie die Benutzer als Spieler ihrer Rollen, mit dem System interagieren.


3.1 Rollen

Die Anwendung unterscheidet zwischen den zwei Rollen Administrator und Feriengast, die jeweils unterschiedliche Handlungen vollziehen können.

Administrator

Als Administrator soll der Benutzer auf Seiten des Urlaubsclubs bezeichnet werden. Er hat also nichts mit einem System-Administrator gemein. In dieser Rolle können diverse Angestellte des Urlaubsclubs vereinigt sein. So könnten der Rezeptionist, der Animateur oder auch der Hotel-Manager in die Lage gelangen die Anwendung in der Rolle des Administrators zu bedienen.

Feriengast

3.2 Use Cases

Abbildung 3.1: Die Anwendungsfälle der Rollen Feriengast und Administrator

**Administrator**

**Einloggen**

Der Administrator muß sich mit seinem Benutzernamen und einem Passwort in die Anwendung einloggen. Erst dann kann er die verfügbaren Dienste nutzen. Dies dient der Sicherheit und dem Mißbrauch der Anwendung durch Feriengäste oder durch Dritte.
Profilvorlage anlegen

Bevor die Feriengäste ihre Profile ausfüllen können, muß der Administrator eine Profilvorlage erstellen. Die Profilvorlage definiert eine bestimmte Art von Profilen. Es sind jeweils beliebig viele Attribute anzugeben (z.B. Tauchen, Tanzen...) und zu jedem Attribut ein zugehöriges Intervall (z.B. 1 bis 10). Dies bedeutet, dass ein Feriengast beim Ausfüllen seines Profils beim Attribut Tauchen Werte zwischen 1 (gar kein Interesse) und 10 (lebensnotwendig) eingeben kann. Die Intervalle können für jedes Attribut einzeln vergeben werden. Folgende Tabelle soll eine mögliche Profilvorlage darstellen:

<table>
<thead>
<tr>
<th>Attribut</th>
<th>untere Grenze</th>
<th>obere Grenze</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tauchen</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Schwimmen</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Tanzen</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Tennis</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Wandern</td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>

Tabelle 3.1: Beispiel einer Profilvorlage

Profile anzeigen lassen

Mit Hilfe dieser Funktionalität kann sich der Administrator alle bereits eingegebenen Profile nebst Namen der zugehörigen Personen anzeigen lassen. Es werden sowohl automatisch erzeugte als auch manuell eingegebene Profile als Tabelle angezeigt. Tabelle 3.2 zeigt ein mögliches Ergebnis dieser Funktion, gemäß der weiter oben beschriebenen Profilvorlage.

<table>
<thead>
<tr>
<th>Name</th>
<th>Tauchen</th>
<th>Schwimmen</th>
<th>Tanzen</th>
<th>Tennis</th>
<th>Wandern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hans Meiser</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Rudi Carrell</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>Kai Pflaume</td>
<td>10</td>
<td>5</td>
<td>1</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

Tabelle 3.2: Beispiel einer Profilliste

Profile in n Cluster ("Tische") einteilen

Hierin besteht die Hauptfunktionalität von get2gether. Die eingegebenen Profile sollen auf Sitzplätze verteilt werden. Das Ziel besteht darin, dass beim Abendessen möglichst jeweils Personen mit ähnlichen Interessen an einem Tisch sitzen, um sich besser kennenlernen und austauschen zu können. Der Administrator kann also eine bestimmte Anzahl von Tischen
vorgeben. Anschließend werden alle Personen - bzw. ihre gesammelten Profile - auf die Anzahl der Tische verteilt.

Schließlich erhält der Benutzer eine Übersicht der Tischeinteilung. Hier ist nun auch eine Auflistung der Profile zu finden, damit sich der Administrator überzeugen kann, dass die Personen an jedem Tisch tatsächlich ähnliche Interessen haben.

**Feriengast**

**Profil ausfüllen**

Der Feriengast geht zu einem der im Club aufgestellten Terminals. Dort gibt er seinen Namen ein und füllt das Profil aus, indem er zu jedem Attribut einen Wert angibt der seine Präferenz dafür widerspiegelt. Vorher muß natürlich von einem Administrator eine Profilvorlage eingegeben worden sein.

**Tischordnung betrachten**

Natürlich möchte der Feriengast zu guter Letzt wissen, an welchem Tisch er beim Abendessen sitzen soll. Hierfür kann er eine Übersicht über die Tischordnung einsehen. Vorher müssen die Profile natürlich von einem Administrator eingeteilt worden sein.

Um keine Datenschutzrichtlinien zu verletzen, sollte dieser Übersicht nicht zu entnehmen sein, welche Werte die einzelnen Profile beinhalten.

### 3.3 Nähe und Distanz

Um die gespeicherten Profile in Gruppen bzw. Tische einteilen zu können, ist es zunächst notwendig zu definieren wann zwei Profile einander ähnlich sind und wie ähnlich sie sich sind. Es wird also ein Maß benötigt, welches z.B. zunehmend niedriger wird, je ähnlicher sich zwei Profile sind und es wird eine Berechnungsvorschrift benötigt um das jeweilige Maß zu bestimmen.

Im Folgenden werden zwei Ansätze vorgestellt, die im Rahmen dieser Arbeit untersucht wurden.
Euklidische Distanz

Die Euklidische Distanz ist eine einfache Möglichkeit um ein Maß für die Ähnlichkeit zweier Profile zu ermitteln. Dabei wird jedes Profil als Punkt im n-dimensionalen Raum gesehen, dessen Koordinaten durch einen Vektor angegeben werden. Die Distanz zwischen zwei Vektoren wird als Länge ("Euklidische Norm" (Net-Lexikon)) des Differenzvektors der beiden Vektoren bestimmt.

Die Euklidische Distanz kann nach folgender Formel berechnet werden (siehe: (Werner, 2004, S. 48)):

\[ |a - b| = \sqrt{\sum_{j=1}^{n} (b_j - a_j)^2} \] (3.1)

Zur besseren Veranschaulichung betrachten wir erneut die Beispiel-Profile aus 3.2. Um die Profile besser darstellen zu können, nehmen wir zudem an, die Profilvorlage bestünde nur aus Tauchen und Schwimmen, so dass wir uns im zweidimensionalen Raum befinden. Wie sehen nun die Maße für die Distanzen der Profile unserer Herren Showmaster aus?

Die folgende Abbildung zeigt ein zweidimensionales Koordinatensystem in dem die Punkte Meiser, Carrell und Pflaume für die Merkmale Tauchen und Schwimmen eingetragen sind:

![Abbildung 3.2: Die Profile der Showmaster als Punkte im Koordinatensystem](image-url)
Es scheint als lägen die Herren Meiser und Carrell recht nah beieinander, wogegen die Herren Meiser und Pflaume einen eher hohen Abstand zueinander aufweisen. Dies soll per Rechnung gezeigt werden:

\[
|\text{Meiser} - \text{Carrell}| = \sqrt{(5 - 3)^2 + (4 - 3)^2} \\
= \sqrt{2^2 + 1^2} \\
= \sqrt{5} \\
= 2,24
\]

\[
|\text{Meiser} - \text{Pflaume}| = \sqrt{(10 - 3)^2 + (5 - 3)^2} \\
= \sqrt{7^2 + 2^2} \\
= \sqrt{53} \\
= 7,28
\]

\[
|\text{Pflaume} - \text{Carrell}| = \sqrt{(5 - 10)^2 + (4 - 5)^2} \\
= \sqrt{-5^2 + (-1)^2} \\
= \sqrt{26} \\
= 5,10
\]

Legt man also die Euklidische Distanz als Maß der Ähnlichkeit zugrunde, ist die Ähnlichkeit zwischen den Herren Meiser und Carrell am Höchsten, während die Interessen der Herren Meiser und Pflaume schon deutlich divergieren.

**Gewichtete Euklidische Distanz**

Nun wäre es vorstellbar, dass in gewissen Anwendungsfällen bestimmte Attribute von Profilen eine größere Rolle spielen sollten als andere.

Eine einfache Möglichkeit dafür ist die Anpassung der euklidischen Distanz. Es kann jedem Attribut ein Gewicht zwischen Null und Eins vergeben werden. Die Summe der Gewichte sollte Eins betragen. Nun kann die Formel zur Berechnung der euklidischen Distanz entsprechend modifiziert werden:

\[
|a - b| = \sqrt{\sum_{j=1}^{n} w_j * (b_j - a_j)^2}
\]  (3.2)
Hierbei entspricht \( w_j \) dem Gewicht für das Attribut mit dem Index \( j \).

Nun können wir wieder das Beispiel von oben aufgreifen. Wir vergeben Gewichte für die Attribute Tauchen und Schwimmen und zwar wie folgt:

\[ w_t = 0,2 \]
\[ w_s = 0,8 \]

Mit diesen Gewichten erhalten wir nun folgende Distanzwerte:

\[
|M - C| = \sqrt{0,2 \times (5 - 3)^2 + 0,8 \times (4 - 3)^2}
= \sqrt{0,2 \times 2^2 + 0,8 \times 1^2}
= \sqrt{0,2 \times 4 + 0,8 \times 1}
= \sqrt{1,6}
= 1,26
\]

\[
|M - P| = \sqrt{0,2 \times (10 - 3)^2 + 0,8 \times (5 - 3)^2}
= \sqrt{0,2 \times 7^2 + 0,8 \times 2^2}
= \sqrt{0,2 \times 49 + 0,8 \times 4}
= \sqrt{13}
= 3,61
\]

\[
|P - C| = \sqrt{0,2 \times (5 - 10)^2 + 0,8 \times (4 - 5)^2}
= \sqrt{0,2 \times (-5)^2 + 0,8 \times (-1)^2}
= \sqrt{0,2 \times 25 + 0,8 \times 1}
= \sqrt{5,8}
= 2,41
\]

Es ist zu beobachten, dass alle drei Distanzen deutlich niedriger geworden sind. So spielt die eigentlich große Distanz von Fünf zwischen den Herren Pflaume und Carrell beim Attribut Tauchen durch die geringe Gewichtung eine viel kleinere Rolle als vorher. Dagegen wird die geringe Distanz von Eins beim Attribut Schwimmen nun durch die hohe Gewichtung deutlicher hervorgehoben, so dass die Gesamtdistanz dieser beiden Profile nun weniger als die Hälfte der ungewichteten euklidischen Distanz beträgt.
Fuzzy-Logik


Die charakteristische Funktion $\mu_F$ ordnet den Elementen einer Menge $X$ einen Wert zwischen Null und eins zu, der den Zugehörigkeitsgrad $r$ der Elemente zu einem unscharfen Begriff darstellt - analog zu (Babic, 2003, S. 79f).

Um im Bild der großen Städte zu bleiben, definieren wir eine Funktion welche Städte mit einer Einwohnerzahl unter 250.000 der Menge der nicht großen Städte zuordnet, während Städte mit mehr als 1 Mio. Einwohnern der Menge der großen Städte zugeordnet werden. Allen Städten deren Einwohnerzahl dazwischen liegt, wird mit Hilfe der Zugehörigkeitsfunktion der Grad $r$ zum Begriff groß zugewiesen:

$$\mu_F(x) = \begin{cases} 
0 & \text{für } 0 \leq x \leq 250000 \\
\frac{4}{300000}x - 333333,3 & \text{für } 250000 \leq x \leq 1000000 \\
1 & \text{für } x > 1000000
\end{cases} \quad (3.3)$$

Zur besseren Veranschaulichung ist in Abb. 3.3 der Graph der Funktion dargestellt.

Laut dieser Funktion ist Frankfurt am Main mit etwa 640.000 Einwohnern mit einer Zugehörigkeit von 0,52 eine große Stadt.

Es lassen sich auf Fuzzy-Mengen u. a. die gleichen Mengenoperationen (Durchschnitt, Vereinigung usw.) anwenden wie auf Mengen der klassischen Mengenlehre. Hier sei aber auf die einschlägige Literatur wie (Görz u. a., 2003, S. 302ff) oder (Lämmel und Cleve, 2001, S. 87) verwiesen.

\textsuperscript{1}Engl.: fuzzy = unscharf, verschwommen
Mit Hilfe der Fuzzy-Logik ist nun eine raffiniertere Berechnung der Nähe bzw. Distanz zweier Profile zueinander denkbar, als es die Euklidische Distanz ermöglicht.

So wäre eine Zugehörigkeitsfunktion denkbar, welche für alle Merkmale eines Profils die Nähe der Werte definiert. Die Funktion könnte für jedes Merkmal gesondert betrachtet werden, so dass z. B. bei manchen Merkmalen die gleiche Distanz mehr wiegt als bei anderen Merkmalen.

Des Weiteren könnte je nach Anwendungsfall die Zugehörigkeitsfunktion für das Merkmal Geschlecht variieren. Im Urlaubsclub für Singles könnte die Zugehörigkeitsfunktion beim Vergleich eines Profils eines Mannes mit dem Profil einer Frau eine Eins ergeben. Wohingegen die Funktion in einem Urlaubsclub für Burschenschaften\(^2\) wohl eher bei zwei männlichen Profilen eine Zugehörigkeit von Eins ergeben würde.

\(^2\)Die Idee ist zugegebenermaßen recht abwegig
3.4 Optimierungsproblem


Reihenfolgeproblem

Bei dem bekanntesten Reihenfolgeproblem handelt es sich um das Travelling Salesman Problem (TSP). Hierbei gilt es für einen Handlungsreisenden die kürzeste Route durch $n$ Städte zu finden, wobei jede Stadt nur einmal besucht werden darf und der Reisende am Ende wieder in seiner Ausgangsstadt ankommen soll.

Das Problem der Verteilung der Personen auf die Tische lässt sich in ähnlicher Form formulieren. Die einzelnen Profile können als $n$ Städte betrachtet werden. Nun will ein Reisender $m$ Reisen durch diese Städte unternehmen. Er will bei jeder Reise nur neue Städte besuchen, die er auf den vorherigen Reisen noch nicht gesehen hat. Es gilt nun für jede einzelne Reise die Strecke zu minimieren - d. h. die "Entfernung" der am gleichen Tisch sitzenden Personen wird minimiert.

Hopfield Netze


Hopfield-Netze beinhalten eine Energiefunktion $E$ welche sich bei jedem Arbeitsschritt verringert. $E$ muß ein Minimum erreichen, sobald sich das Netz im Zustand $n$ befindet.
Nun kann die Energiefunktion auf das jeweils zu lösende Problem angepaßt werden. Will man das TSP mit einem Hopfield-Netz modellieren, so läßt sich die Strecke Hamburg, Berlin, München, Köln und Bremen in dieser Reihenfolge durch ein Netz mit folgender Aktivierung darstellen (eine Eins kennzeichnet jeweils ein aktiviertes Neuron):

<table>
<thead>
<tr>
<th>Stadt/Reihenfolge</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hamburg</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bremen</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Köln</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>München</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Berlin</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Tabelle 3.3: Matrix der Strecke Hamburg, Berlin, München, Köln, Bremen

Die Energiefunktion $E$ die mit Hilfe dieses Netzes minimiert werden soll, muß folgende Bedingungen erfüllen (Zell, 2003, S. 203)

1. $E$ darf nur minimal sein für die Lösungen, die genau eine Eins in jeder Zeile und Spalte haben.

2. $E$ muß für Lösungen mit kürzerer Pfadlänge geringer sein als für solche mit längerer Pfadlänge.

Um diese Bedingungen zu erfüllen, kann eine Funktion definiert werden die aus vier Termen besteht. Die genaue mathematische Formulierung der Termen kann bei Interesse in (Zell, 2003, S. 203) nachgeschlagen werden. Hier soll nur auf die Semantik der Termen eingegangen werden:

- der erste Term darf nur dann Null ergeben, wenn jede Zeile eine Eins enthält. Damit wird sichergestellt, dass jede Stadt nur genau einmal besucht wird.
- der zweite Term darf nur Null ergeben, wenn in jeder Spalte genau eine Eins vorkommt. Dies stellt sicher, dass nicht mehrere Städte gleichzeitig besucht werden können.
- der dritte Term darf nur dann Null ergeben, wenn genau $n$ (mit $n =$ Anzahl der Städte) Einsen in der Matrix vorkommen. Dies stellt sicher, dass auch wirklich jede Stadt besucht wird.
- der vierte Term gibt die Länge der aktivierten Strecke an.

Auf ähnliche Weise, läßt sich nun das „Tischproblem“ mit Hilfe eines Hopfield Netzes modellieren. Der Benutzer müßte vorgeben, auf wie viele Tische die Profile zu verteilen sind. Die mögliche Matrix für drei Tische ist in Tabelle 3.4 dargestellt.

Auch für diese Problemstellung läßt sich nun eine Energiefunktion $E'$ definieren, die aus mehreren Termen besteht:
Tabelle 3.4: Matrix der Verteilung von Showmastern auf Tische

<table>
<thead>
<tr>
<th>Profil/Tischnummer</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kai Pflaume</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rudi Carrell</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Hans Meiser</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Thomas Gottschalk</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Günther Jauch</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ulrich Meyer</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Frank Elstner</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Peter Bond</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- der erste Term darf nur dann Null ergeben, wenn jede Zeile eine 1 enthält. Damit wird sichergestellt, dass jede Person an genau einem Tisch sitzt.

- der zweite Term darf nur Null ergeben, wenn in jeder Spalte mindestens zwei Einsen vorkommen. Dies stellt sicher, dass an jedem Tisch mindestens zwei Personen sitzen (es wäre natürlich auch eine Person vorstellbar).

- der dritte Term darf nur dann Null ergeben, wenn genau \( n \) (mit \( n \) = Anzahl der Profile) Einsen in der Matrix vorkommen. Dies stellt sicher, dass auch wirklich jede Person einen Sitzplatz hat.

- der vierte Term gibt die addierten Distanzen der an jedem Tisch versammelten Profile an.

Clusteranalyse


Mit der Clusteranalyse ließen sich die Profile also mit relativ einfachen Mitteln in Gruppen ein-teilen. Nachfolgend soll zum besseren Verständnis der k-means-Algorithmus (Gerken, 2003, S. 18) erläutert werden, bei dem es sich um ein beliebtes Clustering-Verfahren handelt.
k-means

Um den k-means-Algorithmus anzuwenden, ist folgende Ausgangssituation nötig:

- es existieren $n$ Objekte (im Beispiel die Profile)
- es wird eine Anzahl $k$ der zu generierenden Cluster angegeben
- es existiert eine Abstandsfunktion (z. B. eine der Funktionen aus Abschnitt 3.3)
- es existiert eine Funktion zur Berechnung des Mittelpunktes eines Clusters

Anschließend verläuft der Algorithmus in folgenden Schritten:

1. es werden zufällig $k$ Punkte als Clusterzentren bestimmt
2. jedes Objekt wird dem nächstgelegenen Zentrum zugeordnet
3. für jedes Cluster ist das Zentrum neu zu berechnen
4. Abbruch, falls sich die Zentren nicht mehr ändern; sonst wieder zu Schritt 2

Der k-means-Algorithmus

Fazit

Die Clusteranalyse mit dem k-means-Algorithmus ist mit geringem Aufwand umsetzbar und kann durch Definition einer individuellen Abstandsfunktion an verschiedene Anforderungen angepasst werden. Wohingegen sich der Ansatz, die Profile mit Hilfe von Hopfield-Netzen auf Tische zu verteilen, als kompliziert und aufwändig erweist. Somit verstößt er gegen das gut bewährte KISS-Prinzip\(^3\). Es wird also der Lösungsansatz mit Hilfe der Clusteranalyse bevorzugt.

\(^3\)"Keep it simple, Stupid!"
4 Design und Implementierung

In diesem Abschnitt soll der Entwurf von get2gether dargestellt werden. Dabei wird zunächst von der groben Architektur des Systems ausgegangen, um in den weiteren Abschnitten in die Tiefe zu gehen und die Komponenten sowie das Klassenmodell vorzustellen.

4.1 Systemarchitektur


Java

Aufgrund ihrer Plattformunabhängigkeit bietet sich die Sprache Java zur Implementierung von get2gether an. Dabei bleibt die Unabhängigkeit von der Hardware- und Betriebssystemplattform des Servers gewahrt. Es könnte sich folglich sowohl um einen Windows-Server als auch um einen beliebigen Unix-Server handeln. Des Weiteren erhöht diese Entscheidung die Flexibilität bei der Auswahl der Clients. So könnte z.B. eine Schnittstelle angeboten werden, mit deren Hilfe die Dienste des Servers durch ein Applet per RMI\(^1\) aufgerufen werden könnten. Es wäre auch ein Handy als Client vorstellbar, auf dem die Clientsoftware mit Hilfe eines Midlets\(^2\) realisiert ist.

\(^1\)Remote Method Invocation
\(^2\)Eine Anwendung für Mobiltelefone die dem MIDP-Standard entspricht
Um den eigenen Entwicklungsaufwand möglichst gering zu halten\(^3\), sollen zudem einige Open-Source-Produkte im Rahmen dieser Arbeit genutzt werden. Da besagte Produkte - v. a. im Rahmen des Jakarta-Projektes von Apache\(^4\) - Java als Entwicklungssprache nutzen, bietet es sich auch aus dieser Sicht heraus an.

**Model-View-Controller**


1. Es wird ein HTML-Formular abgeschickt. Der Request wird an das *ActionServlet* geleitet, welches von Struts zur Verfügung gestellt wird.
2. Das *ActionServlet* "sieht" in der Datei *struts-config.xml* nach, welche Aktion für den entsprechenden Request auszuführen ist.
3. Es wird die entsprechende Unterklasse der Struts-Klasse *Action* mit ihrer *execute*-Methode aufgerufen (in der Abbildung *DoSomethingAction*).
5. Die *Action* teilt dem *Servlet* mit was es an den Browser zurückgeben soll (z. B. bei erfolgreichem Administrator-Login eine Admin-Seite und sonst eine Fehlerseite).
6. Das *ActionServlet* sendet die entsprechende Seite zum Browser.

---

\(^3\)"Gute Informatiker sind faule Informatiker" (Prof. Dr. Guido Pfeiffer)
\(^4\)http://jakarta.apache.org
\(^5\)http://struts.apache.org/
4.2 Komponenten


\textsuperscript{6}Java Server Pages
4.3 Klassenmodelle

Web-Komponente


Abbildung 4.2: Die Action-Klassen der Web-Komponente
**Komponente Profilverwaltung**

Die Komponente Profilverwaltung, wie sie in Abb. 4.3 als Klassenmodell dargestellt ist, besteht im Wesentlichen aus den drei Klassen ProfileManager, ProfileTemplate und Profile. Mit Hilfe der Klasse ProfileManager lassen sich u. a. Profilvorlagen definieren, wie dies in 3.2 bereits geschildert wurde. Des Weiteren nimmt diese Klasse alle eingegebenen Profile entgegen und verwaltet diese in einer Liste. Es lassen sich auch Zufallsprofile erzeugen, was der Demonstration von get2gether dienen soll.


**Clustering-Komponente**


Durch die Definition der beiden Interfaces sind verschiedenste Implementierungen von Distanzfunktionen und von Clusterern denkbar. Es wird ein flexibles Modell erreicht, in dem die Algorithmen leicht ausgetauscht und an neue Bedürfnisse angepasst werden können.

---

⁷Im Javajargon ein Objekt vom Typ Map
Konkret wurde für jedes Interface je eine Implementierung realisiert. Die Klasse \textit{WeightedEuklidDistanceFunction} berechnet die Distanz als gewichtete Euklid-Distanz, wie dies bereits in Abschnitt 3.3 beschrieben wurde.

Die Klasse \textit{KMeansClusterer} implementiert den k-means-Algorithmus, wie er in Abschnitt 3.4 erläutert wurde. Dabei benutzt der Algorithmus die gewichtete Euklid-Distanzfunktion um den jeweiligen Abstand zwischen zwei Profilen zu berechnen.

Abbildung 4.3: Klassenmodell der Komponente Profilverwaltung

\(^8\)http://www.cs.waikato.ac.nz/ml/weka/
5 Fallstudie

In diesem Kapitel soll die Anwendung get2gether anhand einer Fallstudie mit Hilfe von Screenshots näher vorgestellt werden.

5.1 Startseite

Zunächst befinden wir uns auf der Startseite (s. Abb. 5.1). In dieser Phase könnte ein möglicher Benutzer sein Profil noch nicht eingeben, da bislang keine Profilvorlage erstellt wurde. Um dies zu tun, loggen wir uns als Admin ein.

5.2 Administrationsmenü

Es erscheint die Administrationsseite, wie in Abb. 5.2 zu sehen. An dieser Stelle hat man die Möglichkeit eine Profilvorlage zu erstellen und sie sich anzeigen zu lassen, Zufallsprofile zu Demonstrationszwecken erzeugen und sich alle vorhandenen Profile anzeigen zu lassen oder die Profile in Cluster einteilen zu lassen. Wir wollen zunächst eine Profilvorlage mit fünf Attributen anlegen.

5.3 Profilvorlage

Es erscheint eine Seite (s. Abb. 5.3) auf der fünf Attribute mit Wertebereich und Gewichtung angegeben werden können. Wir füllen also fünf Attribute aus. Der Wertebereich liegt jeweils zwischen Eins und Zehn. Da uns besonders am Herzen liegt, was die Clubgäste von Tennis halten, gewichten wir dieses Attribut besonders schwer und zwar mit 60%. Alle anderen Attribute erhalten eine Gewichtung von 10%.
5.4 Profil ausfüllen

Nachdem die Profilvorlage erstellt ist, können wir nun Profile eingeben. Dazu loggen wir uns als Admin aus und klicken auf der Startseite auf den Link “Profil eingeben”. Es erscheint die Eingabeseite wie in Abb. 5.4. Wir füllen also ein Profil aus und speichern es.
5.5 Zufallsprofile erzeugen und alle Profile anzeigen

Nun loggen wir uns wieder als Admin ein. Wir lassen uns noch 20 weitere Profile automatisch erzeugen. Diese Funktionalität ist nicht für den praktischen Einsatz gedacht und wurde deshalb nicht in Abschnitt 3.2 erwähnt. Sie dient vielmehr der Demonstration der Anwendung. So müssen nicht diverse Profile von Hand eingegeben werden. Der Admin gibt also die Anzahl der gewünschten Profile an, und prompt werden ebenso viele Profile mit einem
Zufallsnamen und einem zu jedem Attribut passenden Zufallswert erzeugt. Diese Profile sind natürlich gültig gemäß der vorher erstellten Profilvorlage.

Anschließend lassen wir uns alle Profile anzeigen, wie in Abb. 5.5 zu sehen. Unser eingegebenes Profil ist an erster Stelle zu sehen. Es folgen die 20 generierten Profile mit zufälligen Namen und zufälligen Attributwerten, die jeweils im zulässigen Wertebereich liegen.
5.6 Profile in Cluster einteilen

Zu guter Letzt wollen wir uns alle Profile in Cluster einteilen lassen. Wir geben an, dass die Profile auf vier Cluster zu verteilen sind. Anschließend sehen wir - wie in Abb. 5.6 - eine Übersicht aller Cluster und der ihnen zugeordneten Profile. Dabei ist schon beim flüchtigen Blick erkennbar, dass innerhalb jedes Clusters die Werte für das Attribut Tennis deutlich enger zusammenliegen als bei den anderen Attributen.
### Diese Profile wurden bislang eingegeben:

<table>
<thead>
<tr>
<th>Name</th>
<th>wandern</th>
<th>tennis</th>
<th>tanzen</th>
<th>tauchen</th>
<th>schwimmen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kai Praume</td>
<td>3,00</td>
<td>7,00</td>
<td>5,00</td>
<td>6,00</td>
<td>2,00</td>
</tr>
<tr>
<td>KSITGSus0</td>
<td>7,17</td>
<td>5,64</td>
<td>6,76</td>
<td>2,79</td>
<td>9,40</td>
</tr>
<tr>
<td>bjDwUl0f1</td>
<td>6,64</td>
<td>8,94</td>
<td>9,24</td>
<td>2,25</td>
<td>1,70</td>
</tr>
<tr>
<td>IRZAcqua2</td>
<td>5,95</td>
<td>1,00</td>
<td>2,67</td>
<td>4,02</td>
<td>5,27</td>
</tr>
<tr>
<td>Vhgedpqyp3</td>
<td>3,77</td>
<td>6,50</td>
<td>1,12</td>
<td>1,82</td>
<td>1,71</td>
</tr>
<tr>
<td>DMkSvdWcL4</td>
<td>7,86</td>
<td>1,93</td>
<td>9,82</td>
<td>1,94</td>
<td>4,63</td>
</tr>
<tr>
<td>IQSetJXw5</td>
<td>3,29</td>
<td>2,40</td>
<td>4,65</td>
<td>5,57</td>
<td>1,31</td>
</tr>
<tr>
<td>iyAermimU6</td>
<td>5,67</td>
<td>2,42</td>
<td>7,46</td>
<td>7,45</td>
<td>9,49</td>
</tr>
<tr>
<td>pqpusRLyx7</td>
<td>5,62</td>
<td>5,15</td>
<td>8,94</td>
<td>2,38</td>
<td>6,02</td>
</tr>
<tr>
<td>CBqIRwCJ58</td>
<td>6,68</td>
<td>6,48</td>
<td>2,95</td>
<td>4,65</td>
<td>7,73</td>
</tr>
<tr>
<td>kmMbgmSYD9</td>
<td>9,84</td>
<td>4,27</td>
<td>5,05</td>
<td>1,33</td>
<td>1,01</td>
</tr>
<tr>
<td>YVMotQsbw10</td>
<td>7,93</td>
<td>7,93</td>
<td>4,96</td>
<td>2,54</td>
<td>6,47</td>
</tr>
<tr>
<td>KotknawkG11</td>
<td>8,84</td>
<td>9,22</td>
<td>6,70</td>
<td>2,59</td>
<td>7,76</td>
</tr>
<tr>
<td>bqovzFUn12</td>
<td>8,75</td>
<td>5,95</td>
<td>3,54</td>
<td>5,33</td>
<td>1,55</td>
</tr>
<tr>
<td>IMZseDScr13</td>
<td>7,74</td>
<td>4,86</td>
<td>4,63</td>
<td>2,29</td>
<td>8,35</td>
</tr>
<tr>
<td>UTFvDHO14</td>
<td>9,55</td>
<td>6,49</td>
<td>9,98</td>
<td>4,77</td>
<td>2,09</td>
</tr>
<tr>
<td>BnAvZmFr15</td>
<td>4,70</td>
<td>5,65</td>
<td>1,41</td>
<td>8,26</td>
<td>2,95</td>
</tr>
<tr>
<td>iCicBurhN16</td>
<td>6,33</td>
<td>3,89</td>
<td>5,99</td>
<td>9,19</td>
<td>6,14</td>
</tr>
<tr>
<td>rtKZkFZJ17</td>
<td>3,19</td>
<td>5,57</td>
<td>3,34</td>
<td>3,81</td>
<td>1,30</td>
</tr>
<tr>
<td>NRuWMTrw18</td>
<td>3,39</td>
<td>3,53</td>
<td>5,15</td>
<td>2,08</td>
<td>8,01</td>
</tr>
<tr>
<td>EAzhFamJ19</td>
<td>9,77</td>
<td>5,21</td>
<td>6,83</td>
<td>7,98</td>
<td>5,54</td>
</tr>
</tbody>
</table>

Abbildung 5.5: Übersicht aller eingegebenen Profile
Es wurden folgende Cluster erzeugt:

<table>
<thead>
<tr>
<th>Name</th>
<th>wandern</th>
<th>tennis</th>
<th>tanzen</th>
<th>tauchen</th>
<th>schwimmen</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6.64</td>
<td>6.94</td>
<td>9.24</td>
<td>2.25</td>
<td>1.70</td>
</tr>
<tr>
<td>b3dWUUDf1</td>
<td>7.93</td>
<td>7.93</td>
<td>4.96</td>
<td>2.54</td>
<td>6.47</td>
</tr>
<tr>
<td>YVMotOsbw10</td>
<td>8.04</td>
<td>9.22</td>
<td>6.70</td>
<td>2.59</td>
<td>7.76</td>
</tr>
<tr>
<td>KotknawgG11</td>
<td>9.55</td>
<td>6.49</td>
<td>9.98</td>
<td>4.77</td>
<td>2.03</td>
</tr>
<tr>
<td>1</td>
<td>7.17</td>
<td>5.54</td>
<td>6.76</td>
<td>2.79</td>
<td>9.40</td>
</tr>
<tr>
<td>pqpusPlvy7</td>
<td>5.62</td>
<td>5.15</td>
<td>8.94</td>
<td>2.38</td>
<td>5.02</td>
</tr>
<tr>
<td>CBqlRWCJ58</td>
<td>6.68</td>
<td>6.48</td>
<td>2.95</td>
<td>4.55</td>
<td>7.73</td>
</tr>
<tr>
<td>kmMbgmSY09</td>
<td>9.84</td>
<td>4.27</td>
<td>5.85</td>
<td>1.33</td>
<td>1.01</td>
</tr>
<tr>
<td>IMZzLDzcr13</td>
<td>7.74</td>
<td>4.86</td>
<td>4.63</td>
<td>2.29</td>
<td>8.35</td>
</tr>
<tr>
<td>EAzHSaMj19</td>
<td>9.77</td>
<td>5.21</td>
<td>6.83</td>
<td>7.98</td>
<td>5.54</td>
</tr>
<tr>
<td>2</td>
<td>5.95</td>
<td>1.00</td>
<td>2.67</td>
<td>4.02</td>
<td>5.27</td>
</tr>
<tr>
<td>IRZAcqua2</td>
<td>7.86</td>
<td>1.96</td>
<td>9.92</td>
<td>1.94</td>
<td>4.63</td>
</tr>
<tr>
<td>DMkSvdWcL4</td>
<td>3.29</td>
<td>2.40</td>
<td>4.65</td>
<td>5.57</td>
<td>1.31</td>
</tr>
<tr>
<td>kQSetJXW5</td>
<td>5.67</td>
<td>2.42</td>
<td>7.46</td>
<td>4.93</td>
<td>9.49</td>
</tr>
<tr>
<td>iyAemnMU6</td>
<td>6.33</td>
<td>3.96</td>
<td>5.99</td>
<td>9.18</td>
<td>5.14</td>
</tr>
<tr>
<td>iCicBuh-N15</td>
<td>3.39</td>
<td>3.58</td>
<td>5.15</td>
<td>2.08</td>
<td>8.01</td>
</tr>
<tr>
<td>3</td>
<td>3.00</td>
<td>7.00</td>
<td>5.00</td>
<td>5.00</td>
<td>2.00</td>
</tr>
<tr>
<td>Yhgepdqyp3</td>
<td>3.77</td>
<td>6.50</td>
<td>1.12</td>
<td>1.82</td>
<td>1.71</td>
</tr>
<tr>
<td>bqovZFMN12</td>
<td>8.75</td>
<td>5.96</td>
<td>3.54</td>
<td>5.33</td>
<td>1.55</td>
</tr>
<tr>
<td>BnazzNaeF15</td>
<td>4.70</td>
<td>5.66</td>
<td>1.41</td>
<td>8.28</td>
<td>2.95</td>
</tr>
<tr>
<td>rtKZKFzJ17</td>
<td>3.19</td>
<td>5.57</td>
<td>3.34</td>
<td>3.81</td>
<td>1.30</td>
</tr>
</tbody>
</table>

Abbildung 5.6: Übersicht aller geclusterter Profile
6 Ausblick

In diesem Abschnitt soll auf einige Punkte eingegangen werden, die in dieser Version von get2gether nicht umgesetzt wurden, die jedoch in eine "produktreife" Version Eingang finden müßten.

6.1 PDA

Für den endgültigen Einsatz in einem Ferienclub sollte die Anwendung auf PDAs an die Feriengäste verteilt werden, wie dies im Szenario in Abschnitt 2.1 vorgestellt wurde. Aufgrund der Architektur von get2gether, das als Client-Server-Anwendung unter Einsatz des MVC-Paradigmas realisiert ist, ließe sich relativ einfach eine alternative Oberfläche umsetzen.

6.2 Sicherheit


Des weiteren sollte der Datenverkehr zwischen Client und Server u. U. verschlüsselt erfolgen, damit er nicht abgehört werden kann. Hierzu bietet sich das Protokoll HTTPS\(^1\) an, bei dem die Verbindung zwischen Client und Server verschlüsselt wird.

\(^1\)Hypertext Transfer Protocol Secure
6 Ausblick

6.3 Persistenz

Literaturverzeichnis


lexikon-definition.de/Euklidische-Distanz.html. – Zugriffsdatum: 2004-09-26

[Werner 2004] WERNER, Bodo: Mathematik für das Lehramt an der Grund- und Mittel-
stufe sowie an Sonderschulen Teil II (Lineare Algebra und analytische Geometrie): SoSe
werner/GruMiSoSe04TeilII.pdf. – Zugriffsdatum: 2004-09-28


S. 338–353

[Zell 2003] ZELL, Andreas: Simulation neuronaler Netze. 4. Auflage. Oldenbourg Wissen-
schaftsverlag, 2003