

**JohannWolfgangGoetheUniversität
FrankfurtamMain**

ProseminarSS 2000

DynamicQueries Steuerelemente

**Fabian Wleklinski
(fabian@wleklinski.de)**

13. August 2000

1 Inhaltsverzeichnis

1	<u>INHALTSVERZEICHNIS</u>	2
2	<u>VORWORT</u>	4
3	<u>EINBEISPIELZURMOTIVATION: DYNAMIC HOMEFINDER</u>	5
3.1	EINLEITUNG	5
3.2	STEUERELEMENTESTATT SQL	5
3.3	ÜBERGANGZUR PROBLEMSTELLUNG	6
4	<u>STEUERELEMENTE</u>	8
4.1	AUFGABENDER STEUERELEMENTE	8
4.2	KRITERIENFÜR STEUERELEMENTE	9
5	<u>DIEPROBLEMSTELLUNG: KLASISCHESTEUERELEMENTE</u>	10
5.1	EINLEITUNG	10
5.2	DER SLIDER (SCHIEBEREGLER)	10
5.3	DIE CHECKBOX (KONTROLLKÄSTCHEN)	11
5.4	DER RADIOPUSHBUTTON (OPTIONSFELD)	12
5.5	DIE COMBOBOX (KOMBINATIONSFELD)	12
5.6	ZUSAMMENFASSUNG	13
6	<u>DIELÖSUNG(1):NEUARTIGESTEUERELEMENTE</u>	14
6.1	DAS STEUERELEMENT “ALPHASLIDER“	14
6.2	DAS STEUERELEMENT “DATA VISUALIZATION SLIDER”	16
6.3	DAS STEUERELEMENT “2DWIDGET“	19
6.4	ZUSAMMENFASSUNG	20
7	<u>DIELÖSUNG(2):MAGIC LENSES</u>	21
7.1	EINLEITUNG	21
7.2	LINSENALS FILTER	21
7.3	DETAILS ON DEMAND	22
7.4	VERWENDUNG VON MEHRERER LINSE	23
7.5	GRUPPIERUNG	24
7.6	FUZZY LOGIC	24
7.7	MISSING VALUES	24
8	<u>ZUSAMMENFASSUNGUND AUSBLICK</u>	26

9 KOMMENTIERTESLITERATURVERZEICHNIS

27

10 ABBILDUNGSVERZEICHNIS

29

11 INDEX

30

2 Vorwort

Das Aussterben nicht -visueller Anwendungen für den gemeinsamen Anwender ist allg. gegenwärtig, und macht auch vor der Datenbankabfrage nichthalt. Die technologisch enFor tschritte der neunziger Jahre in Bezug auf Massenspeicher und Netzwerke haben zu einem Overkill an Daten geführt, der mit den kümmerlichen Datenbanken fragen nur noch schwer zu bewältigen ist.

DynamicQueries verspricht eine Lösung für dieses Problem: kryptische SQL -Ausdrücke und Kommandozeilen-Werkzeug gewechseln der Echtzeit -Simulation in der virtuellen Realität.

Eine Datenbankabfrage wird nicht mehr erstartet, und läuft bis zu ihrem „Ende“, sondern ist vielmehr eine kontinuierliche Interaktion zwischen dem Anwender und der Anwendung. Im Sinne dessen, so genannten, „Tight Coupling“ agieren und reagieren die Anwendung und der Anwender gegenseitig auf ihre Aktionen. Eine ausgefeilte Oberfläche vermittelt dem Anwender dabei das Gefühl, „live“ ander Abfrage-Teile zu haben.

Die Anwendung greift die eigentlichen Datenbankabfragen voraus, indem Steuerelemente ihren Wert eingeben so anpassen, dass keine unsinnigen Abfragen möglich sind. Umgekehrt kann das Abfrageergebnis neutral als Abfrage verwendet werden. Durch Mausklicks werden Abfragen kontinuierlich präzisiert und neu formuliert, der Abfragevorgang bekommt einen, bzw. wserartigen Charakter.

DynamicQueries bedeutet neue Anforderungen an grafische Ressourcen, Algorithmen, Datenbankstrukturen, Schnittstellen und Protokolle, sowie an die verwendeten Steuerelemente. Dieser letzte genannte Aspekt ist der Inhalt dieses Papiers:

Die Herausforderungen für Steuerelemente durch *DynamicQueries*.

3 Ein Beispiel zur Motivation: *Dynamic HomeFinder*

3.1 Einleitung

Ben Shneiderman unterrichtet seit mehr als 16 Jahren an der *University of Maryland*, und gehört zu den Pionieren der *Dynamic Queries*. Anfang der neunziger Jahre entwickelte er in Schülern Christopher Williamson die Anwendung „*Dynamic HomeFinder*“, um die Wirkung von *Dynamic Queries* durch „*Tight Coupling*“ auf Anwender zu demonstrieren, und zu erforschen.

Im Jahr 1994 gründet Christopher Williamson zusammen mit seiner Frau Cynthia Williamson die Firma „*DreamQuest Software*“ (<http://www.dqsoft.com>), die Computer Spiele entwickelt und vertreibt.

Im Jahr 1995 entwickelt Christopher Williamson zusammen mit Tom Smallwood eine erste Version des *Dynamic HomeFinder*, als Abschlussarbeit für den Titel „*Masters of Engineering in Software Engineering*“¹. Diese Software kann unter der URL <http://www.dqsoft.com/homefind> abgerufen werden, und dient im Folgenden dazu, den Bedarf für neuartige Steuerelemente durch *Dynamic Queries* zu demonstrieren.

Der *Dynamic HomeFinder* ist eine Software, die die Suche von Immobilien in einem vorgegebenen Datenbestand erlaubt, und sich dabei auf Mittel von *Dynamic Queries*, insbesondere *Tight Coupling*, bedient.

3.2 SteuerelementestattSQL

Werschone in mal mit einem konventionellen Datenbanksystem gearbeitet, so dass der Kenntnis der SQL-Befehle. Kryptische Kommandos in der Form wie „`select * from homes`“ veranlassen einen Datenbankserver, hunderte oder tausende Zeilen voller Werte auszudrucken.

Wesentlich eleganter geschieht die Datenbankabfrage mit dem *Dynamic HomeFinder*. Nach dem Programmstart präsentiert sich diese Anwendung wie folgt:

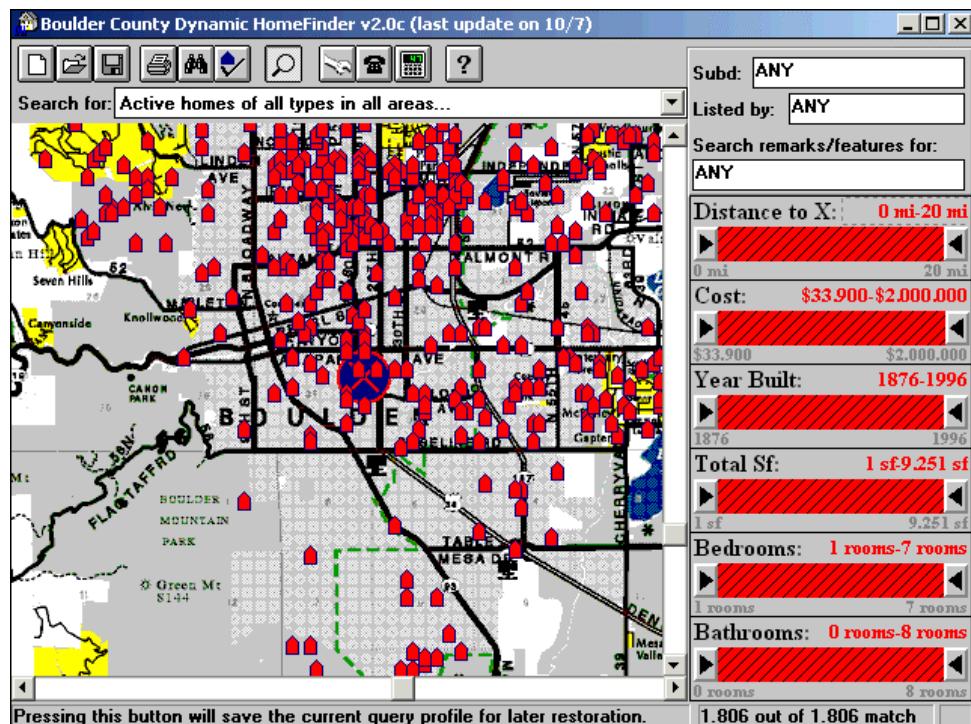


Abbildung 1 – *Dynamic HomeFinder* – `SELECT*FROMHOMES`

¹ Die Abschlussarbeit ist in deutscher Sprache geführt und als Diplom

Im linken Teil des Bildschirmfotos ist ein „Starfield Display“ zusehen, auf welchem jede gefundene Immobilie, d.h. jeder Datensatz, den die Anfrage beschreibt, durch ein kleines Symbol in Form eines Hauses dargestellt ist.

Im rechten Teil des Bildschirmfotos befindet sich die sogenannten „Steuerelemente“. Dass sind die Bestandteile der Oberfläche einer Anwendung, die der Anwender (im Allgemeinen) mit der Maus betätigt (klickt, verschiebt, drückt, ...), um dadurch eine bestimmte Eingabe gemacht. Siehe auch „4 Steuerelemente“.

Durch einfaches und intuitives „Herumdrehen“ und „Ziehen“ können die Steuerelemente kennende Auswahlkriterien für die Datenbankabfrage definiert werden, z.B. Ober- und Untergrenzen für „Anzahl der Badezimmer“, „Erstes Nutzungsjahr“ und „Preis“ einer Immobilie.

Während der Anwender durch das Betätigen der Steuerelemente im rechten Teil des Bildschirms eine individuellen Präferenz definiert, wird die Abfrage gerade in Gestalt der Ansicht im linken Teil des Bildschirms aktualisiert.

Nachdem der Anwender seine Präferenz definiert hat, ist die Menge der gefundenen Immobilien (Datensätze) geringer als zum Programmstart. Die Oberfläche sieht dann z.B. wie folgt aus:

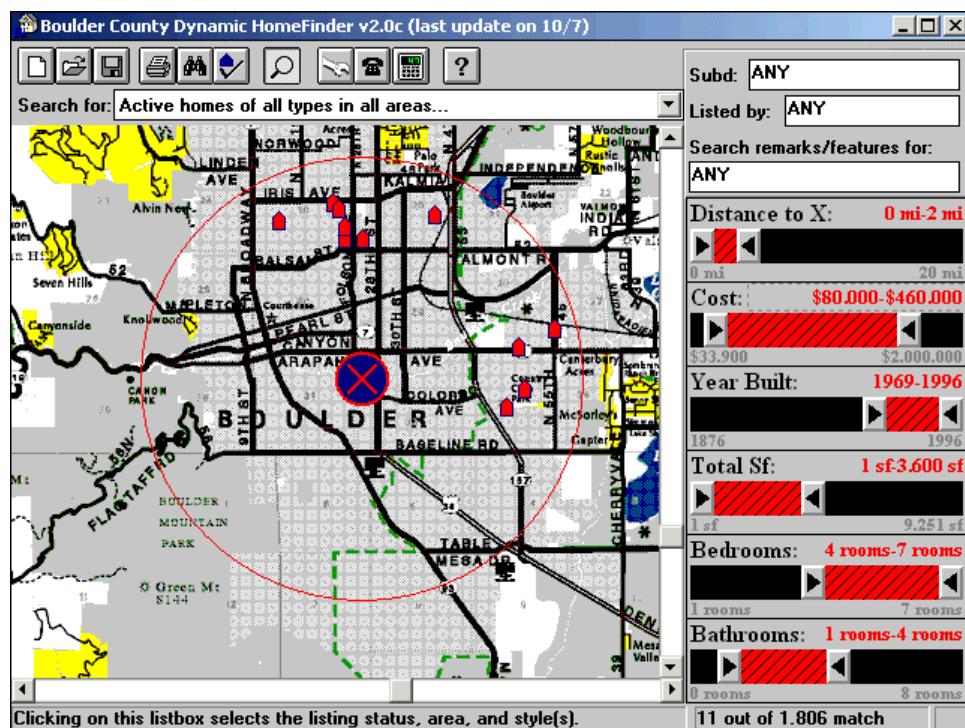


Abbildung 2 – Dynamic HomeFinder während einer typischen Abfrage

Eine solche Abfrage artet ohne die Benutzung von Dynamic Queries schnell in einem Zeilenlangen und unübersichtlichen Konspektwiedem folgendem aus:

```
select * from homes where (cost >= 80000) and (cost <= 460000) and
(year >= 1969) and (year <= 1996) and (bedrooms >= 4) and (bedrooms <= 7)
and (bathrooms >= 1) and (bathrooms <= 4) and (distance <= 2) and (total <= 3600);
```

3.3 Übergang zur Problemstellung

Damit ist nun noch einmal ein Beispiel des Dynamic HomeFinder gezeigt worden, was Dynamic Queries eigentlich sind, und wie man sie anwenden kann. Die Untersuchungen von Ben Shneiderman aber zeigen, dass diese Art von Benutzeroberfläche von der Mehrheit der Anwender akzeptiert wird. Ein Anwender soll sogar gesagt haben: „Ich will nicht aufhören, das macht Spaß!“

DynamicQueries Steuerelemente

Ohnedarauf einzugehen,sind beim *Dynamic HomeFinder* bereitserweiterte Steuerelemente zum Einsatz gekommen. Die verwendeten Schieberegler zur Auswahl von Ober - und Untergrenzen werden nämlich von keiner populären, grafischen Oberfläche wie *MS Windows*, *KDE* oder *Swing* zur Verfügung gestellt, und sind von den allerwenigsten Anwendern schon einmal benutzt worden. Dass aber trotzdem jeder Anwender auf Anhieb mit der Anwendung arbeiten konnte, und sogar Spaß dabei im gefunden hat, verdeutlicht die Wichtigkeit der Entwicklung neuer, und „guter“ Steuerelemente.

4 Steuerelemente

Nachdem in dem vorangegangenen Kapitel festgestellt wurde, was man mit Steuerelementen machen kann, und wie nützlich diese sind, geht es nun darum, Kriterien zu definieren, anhand derer man verschiedene Steuerelemente miteinander vergleichen kann.

Diese Kriterien werden in den folgenden Kapiteln benutzt, um die „Nachteile“ der klassischen, und die „Vorteile“ der neuartigen Steuerelemente herauszustellen.

4.1 Aufgabe der Steuerelemente

Steuerelemente haben im Wesentlichen drei Aufgaben:

1. Definieren einer Auswahl

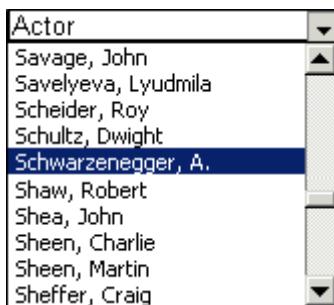


Abbildung 3 - Auswahlaufgabe

Der Anwender definierte eine Auswahl, um der Anwendung etwas „mitzuteilen“.

Dazu gehören z.B. das „Ziehen“ an einem Schieberegler, das „Anklicken“ eines Kontrollkästchens oder das Eingeben von Text in ein Texteingabefeld.

Hier berichtet der Anwender Informationen an die Anwendung weiter.

2. Visualisierung von Informationen

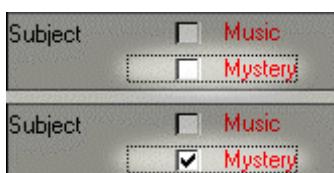


Abbildung 4 - Informationsaufgabe

Nachdem der Anwender eine Auswahl getroffen hat (oder auch schon vorher), wird diese Auswahl weiterhin angezeigt. Wenn beispielsweise ein Schieberegler „gezogen“ wurde, verharrt der Schieberegler in der neuen Position, die der Anwender jederzeit ändern kann.

Alleine das Vorhandensein eines Steuerelementes informiert den Anwender bereits, und zwar über die Existenz dieser Eingabemöglichkeit, und ggf. über den zulässigen Wertebereich, aus dem der Anwender auswählen kann.

Hier berichtet die Anwendung Informationen an die Anwendung weiter.

3. Auslösen von Aktionen

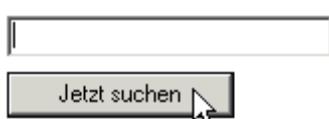


Abbildung 5 - Informationsaufgabe

Die Anwendung reagiert mit der Ausführung von Programmcode auf das Betätigen eines Steuerelementes. Dazu gehört typischerweise das „Klicken“ auf eine Schaltfläche, eine sogenannte „Button“.

Im Sinn des „Tight Coupling“ ist dieser Aspekt eines Steuerelementes für *Dynamic Queries* von untergeordneter Bedeutung, da Aktionen der Anwendung in Echtzeit geschehen, und nicht explizit durch Schaltflächen ausgelöst werden müssen.

² Beim Wertebereich handelt es sich z.B. im Falle eines Kontrollkästchens um „ja“ und „nein“, im Fall eines Texteingabefeldes um alle Texte, usw.

4.2 Kriterien für Steuerelemente

Kriterien für Steuerelemente lassen sich finden wie Sand am Meer. Die Steuerelemente müssen sich in der Handhabung auf eine Handvoll richten, um den Nutzen für *Dynamic Queries* relevant zu machen.

1. Platzbedarf

Jedes Steuerelement benötigt Platz, „Platz“ im Sinn eines Bereiches auf dem Bildschirm, indem das Steuerelement dargestellt wird, und dies darf die unterliegenden Bereiche (indem sie z.B. weitere Steuerelemente befinden können) verdecken. Da die Größe des Bildschirms begrenzt ist, gilt es, mit der Ressource „Platz“ umsichtig zu wirtschaften.

Insbesondere für *Dynamic Queries* ist der Platzbedarf eines Steuerelementes interessant, da es hierbei um die Abbildung einer Menge von Datensätzen auf einen Wertebereich geht. Stehen beispielsweise eine Million verschiedene Datensätze zur Verfügung, so wäre es wenig zweckmäßig, diese Datensätze alle untereinander in einer großen Liste darstellen zu wollen.

2. Diskrete Werte

Je nach Verwendungszweck der Steuerelemente kann es gewünscht sein, dass ausgewählte Werte denken können – oder aber auch nicht. Das „perfekte“ Steuerelement würde natürlich beide Varianten zulassen, und die Auswahl dem Anwender überlassen.

Das „perfekte“ Steuerelement „aber im Moment außer Reichweite ist“, sollte ein Steuerelement vor allem die Auswahl diskreter Werte ermöglichen. Zwar mag z.B. die Auswahl von Größen wie „Lautstärke“ oder „Kaufpreis“ eine Toleranz von x Prozent sinnvoll sein, bei der Auswahl von „Anzahl der Badezimmer“ aber wohl kaum.

Beispiel: Angabe der Anzahl der Badezimmer der gesuchten Immobilie.

3. Bereichsauswahl

Ein Steuerelement sollte die Möglichkeiten anbieten, sofern das im jeweiligen Kontext sinnvoll ist, nicht nur in einzelnen Werten, sondern in einem Bereich auszuwählen. Die Möglichkeit, einen einzelnen Wert auszuwählen, sollte natürlich erhalten bleiben.

Beispiel: Angabe eines Wunschkaufspreises der gesuchten Immobilie mit Unter- und Obergrenze.

4. Mehrfachselektion

Ein Steuerelement sollte die Möglichkeiten anbieten, nicht nur in einzelnen Werten, sondern auch in einem Wertebereich, sondern auch mehrere Werte oder Wertebereiche auszuwählen.

5. Browsen

Für *Dynamic Queries* ist der Charakter des „browsen“ unabdingbar. Unter „browsen“ versteht man eine Art von zielgerichtetem Suchen, z.B. das Variieren eines Schiebereglers.

6. Informationsvisualisierung

Wie in „4.1 Aufgaben der Steuerelemente“ dargelegt, haben Steuerelemente eine Informationsszweck. Verschiedene Steuerelemente unterscheiden sich in dem Umfang und der Qualität, indem sie dem Anwender Informationen zur Verfügung stellen.

7. Bedienungskomfort

Generell soll ein Steuerelement für den Anwender so einfach wie möglich zu bedienen sein. Niedrigerer Bedienkomfort kann den Anwender entmotivieren, verärgern, und ggf. von der Benutzung und/oder dem Kauf abhalten.

Nachdem nun eine Handvoll Kriterien für die Bewertung von Steuerelementen gefunden sind, gilt es im folgenden Kapitel, die klassischen Steuerelemente anhand dieser Kriterien zu bewerten.

5 Die Problemstellung: klassische Steuerelemente

5.1 Einleitung

Jeder kennt Sie, die Schaltflächen „Schieberegler“, Kombinationsfelder und Kontrollkästchen – oder eingedeutscht: „Buttons“, „Slider“, „Comboboxen“ und „Checkboxen“. Sie werden von Benutzern überflächen und Betriebssystemen aller Art zur Verfügung gestellt, als kleine Auswahlseien hier *CDE*, *KDE*, *MS Windows*, *Mac OS*, *Swing*, *Open Look*, *Palm OS* genannt.

Es wird nun gezeigt, dass die klassischen Steuerelemente – trotz ihrer Allgemeinweltigkeit – keine sogenannten Kriteriengerechtwerden, die *Dynamic Queries* anstreben.

5.2 Der Slider (Schieberegler)

Der Slider, oder: Schieberegler ermöglicht das Auswählen eines diskreten Wertes aus einem festvorgegebenen Wertebereich. Ein klassisches Beispiel für die Verwendung des Schiebereglers ist der Einsatz als Lautstärkeregler:



Abbildung 6 – Ein Schieberegler

In Abbildung 6 sind sogar gleich zwei Schieberegler zu sehen, je einer für Lautstärke und Balance. Der Schieberegler besitzt den Vorteil, dass er mit einem konstanten Platz auskommt.

Der Schieberegler besitzt, theoretisch den Vorteil, dass diskrete Werte gewählt werden können. „Theoretisch“ deswegen, weil bei einer bestimmten Größe des Wertebereichs aus technischen Gründen nur noch jeder zweite Wert gewählt werden kann. Im Falle des Lautstärkereglers ist das sicherlich nicht nachteilig, aber dass sieht anders aus, wenn es sich z.B. um den Kaufpreis einer Immobilie handelt.

Das „browsen“ ist mit dem Schieberegler sehr gut möglich, und der Bedienkomfort ist relativ hoch.

Der klassische Schieberegler bietet aber keine Bereichsauswahl und auch keine Mehrfachselektion an. Die Informationsvisualisierung des Schiebereglers ist eher schlecht, denn er teilt dem Anwender nichts über den Wertebereich, den aktuellen Wert, die Werteverteilung oder die gleichen mit, und das, obwohl der Schieberegler über einige „freie Flächen“ verfügt, auf denen Informationen untergebracht werden könnten.

³ Sowohl Maus als auch Bildschirm verfügen über eine bestimmte Auflösungsgrenze, innerhalb derer die Maus keine Bewegung mehr wahrnimmt, bzw. der Bildschirm keine Veränderung mehr darstellen kann.

Daher:

	Platz	Diskrete Werte	Bereich-Auswahl	Mehrfach-Selektion	„browsen“	Visualisierung	Komfort
Slider	+				+		+

5.3 Die Checkbox (Kontrollkästchen)

Das Kontrollkästchen oder die „Checkbox“ ermöglicht eine feste oder eine ausgewählte Option in einer Gruppierung mit anderen Kontrollkästchen auf.



Abbildung 7 - Einige Kontrollkästchen

Der Platzbedarf einer Kontrollkästchen-Gruppe steigt linear mit der Größe des Wertebereiches – was es für eine Verwendung im Bereich der *Dynamic Queries* nur bedingt einsetzbar macht. Mehr als etwa zwanzig, oder auch dreißig verschiedene Optionen sind mit Kontrollkästchen (auch bei „großem“ Bildschirm) nicht visualisierbar. Der Einsatz eines Schiebereglers, um zu erreichen, dass immer nur einige Kontrollkästchen gleichzeitig auf dem Bildschirm zu sehen sind, geht erheblich zu Lasten des Komforts.

Eine Gruppe von Kontrollkästchen erlaubt eine Abhängigkeit von der Größe des Wertebereiches die Selektion und Mehrfachselektion diskreter Werte. Eine Bereichsauswahl ist durch die Mehrfachselektion zwar möglich, aber bei großem Wertebereich sehr unkomfortabel.

Von einem „browsen“ kann hingegen nicht die Redesein. Wenn der Anwender alle Kontrollkästchen gemäß seinen Vorstellungen einstellt, und dann abrupt seine Meinung ändert, sind für ihn bis zu manuelle Arbeitsschritten notwendig. Bei einem Datenbestand von 1 Mio. Datensätzen erfordert das vom Anwender also bis zu 1 Mio. Mausklicks.

Die Informationsvisualisierung ist ausreichend, aber nicht perfekt. Dadurch, dass jede Option separat dargestellt wird, kann der Anwender sehen, welche Optionen zur Verfügung stehen. Es werden ihm aber keine darüber hinausgehenden Informationen angeboten, z.B. muss eine Konfiguration von Kontrollkästchen erstaunlich probieren, um festzustellen, was sich dahinter verbirgt.

Der Bedienungskomfort ist gut, verschlechtert sich aber zunehmend mit der Anzahl der Kontrollkästen.

Daher:

	Platz	Diskrete Werte	Bereich-Auswahl	Mehrfach-Selektion	„browsen“	Visualisierung	Komfort
Checkbox		+	+	+			+

5.4 Der Radiobutton (Optionsfeld)

Das Optionsfeld „meistens „Radiobutton“ genannt, entspricht in wesentlichen Punkten der Checkbox“, siehe „5.3 Die Checkbox (Kontrollkästchen)“:

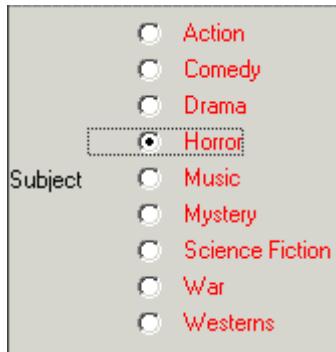


Abbildung 8 - Ein Optionsfeld

Anders als bei dem Kontrollkästchen kann immer nur einer, und nicht beliebig viele Werte ausgewählt werden. Typischerweise wird das Optionsfeld auch aus diesem Grund verwendet: um auszuschließen, dass der Anwender versehentlich mehrere Werte auswählt, obwohl eigentlich nur ein Wert sinnvoll ist.

Das Optionsfeld erfüllt die Kriterien also in gleichem Maße wie das Kontrollkästchen, mit der Ausnahme, dass Mehrfachselektion und Bereichsauswahl nicht möglich sind.

Daher:

	Platz	Diskrete Werte	Bereich-Auswahl	Mehrfach-Selektion	„browsen“	Visualisierung	Komfort
Radiobutton		+					+

5.5 Die Combobox (Kombinationsfeld)

Das Kombinationsfeld ermöglicht, einen freien Text einzugeben, oder einen Text aus einer Liste von vorgegebenen Texten auszuwählen.

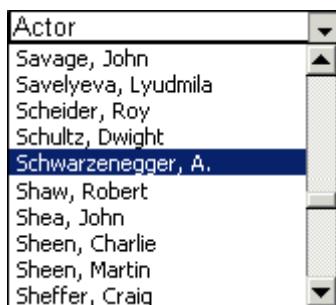


Abbildung 9 - Ein Kombinationsfeld

Der Platzbe darf des Kombinationsfeldes ist konstant. Die Liste, aus welcher der Anwender einen von vorgegebenen Texten auswählt, besteht aus einer Konstanten Höhe, in Abbildung 9 z.B. zehn Zeilen. Existieren mehrere vorgegebene Werte, als die Liste fasst, wird ein zusätzlicher Schieberegler angezeigt, wie in Abbildung 9 zu sehen ist.

Das Kombinationsfeld erlaubt eine Abhängigkeit von der Größe des Wertebereiches, ein endiskreter Wert auszuwählen. Mehrfachselektion und Bereichsauswahl sind nicht möglich.

VoneinemBrowserkannhierebenfallsnichtdieRedesein:DerAufwandfürdenAnwender ,ein ElementinderListe zu „finden“, undanzuwählen, steigtproportionalzurGröße des Wertebereiches an. DaranändertaucdieTatsachenichts, dass dieListelexikographischsortiertist, undineinigenIm p-lementierungensogareinschne llesNavigiereninderListemitdenBuchstabentastenmöglichist ⁴.

Der Bedienungskomfort des Kombinationsfeldes ist relativ niedrig. Gleiches gilt für die Güte der Informationsvisualisierung.

Daher:

	Platz	Diskrete Werte	Bereich-Auswahl	Mehrfach-Selektion	„browsen“	Visualisierung	Komfort
Combobox	+	+					

5.6 Zusammenfassung

Zusammenfassender ergibt sich für die klassischen Steuerelemente das folgende Bild:

	Platz	Diskrete Werte	Bereich-Auswahl	Mehrfach-Selektion	„browsen“	Visualisierung	Komfort
Slider	+				+		+
Checkbox		+	+	+			+
Radiobutton		+					+
Combobox	+	+					

Summa summarum: Es gibt viel zu tun.

⁴ Ein Druck auf „z“ bewirkt dann, dass der erste Eintrag der Liste , der mit „z“ beginnt, ausgewählt wird. Dadurch wird der Suchaufwand in etwa auf 1/26 reduziert.

6 Die Lösung(1):neuartige Steuerelemente

Es sind vorangehenden Kapitel gezeigt worden, dass Steuerelemente für *DynamicQueries* unabdingbar sind, und dass die klassischen Steuerelemente nicht allen Anforderungen gewachsen sind, die *DynamicQueries* ansiestellen. Was wäre eine liegender, als zu versuchen, die klassischen Steuerelemente zu erweitern?

Im Folgenden werden Ansätze von Ben Shneiderman, Christopher Ahlberg und Stephen G. Eick gezeigt, durch die Erfindung neuartiger Steuerelemente die Unzulänglichkeit der klassischen Steuerelemente auszuräumen, und so den Effekt des *Tight Coupling* zu verbessern, oder ihn zu ermöglichen.

6.1 Das Steuerelement "Alphaslider"

Der „Alphaslider“ ist eine von Ben Shneiderman und Christopher Ahlberg entwickelte Erweiterung des klassischen Schiebereglers⁵. Im Gegensatz zu dem klassischen Schieberegler besitzt der Alphaslider einen Textfelder im Wertebereich, unterlaubt optional die Auswahl eines Bereiches anstelle eines diskreten Wertes:

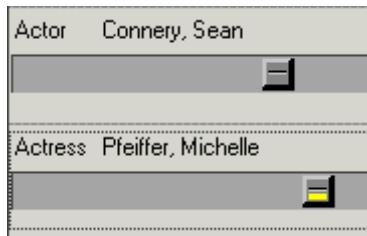


Abbildung 10 – Alphaslider

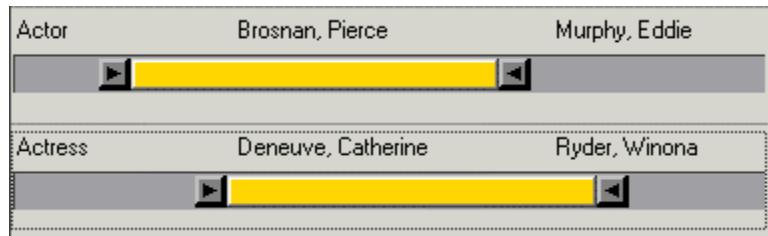


Abbildung 11 - Alphaslider (Bereichsmarkierung)

⁵ Implementierungen finden sich u.a. in Ahlbergs Entwicklungen, „Dynamic HomeFinder“ und „Dynamic FilmFinder“. Die verwendeten Bildschirme fotos entstammen dem „Dynamic FilmFinder“:

Jenach Sinn und Zweck kann der Entwickler einer Anwendung wählen, von Werten oder von Bereichen vorgeben, oder diese Wahl dem Anwender überlassen. Ben Schneider mangelt in seiner Software „Dynamic FilmFinder“ dem Anwender über einen Kontextmenü die Möglichkeit, sich selber für einen Modus zu entscheiden:

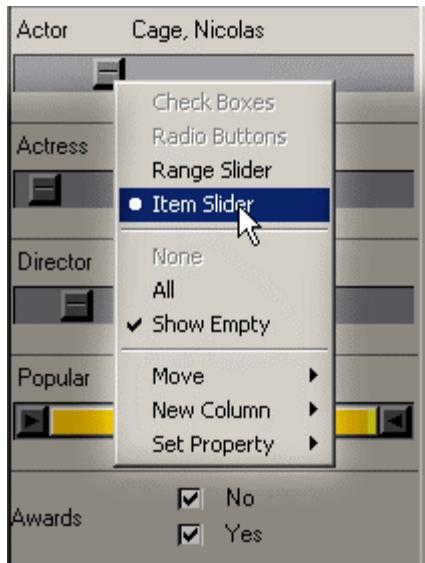


Abbildung 12 - Alphaslider Kontextmenü1



Abbildung 13 - Alphaslider Kontextmenü2

Der Alphaslider von Ben Schneider man verfügt zusätzlich über fortgeschrittene Visualisierungsfunktionen. In Abbildung 14 und Abbildung 15 ist zu sehen, wie sich das Aussehen des Alphasliders in Abhängigkeit der Existenz von Datensätzen für die jeweilige Einstellung verändert⁶:

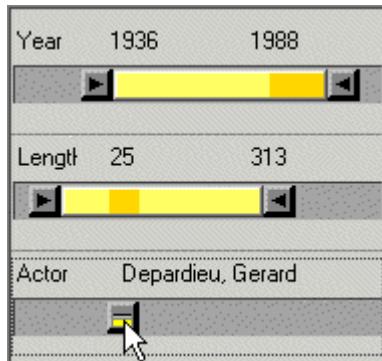


Abbildung 14 - Alphaslider Visualisierung1

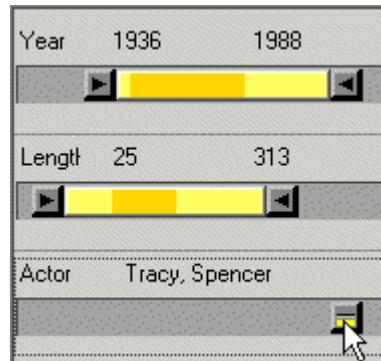


Abbildung 15 - Alphaslider Visualisierung2

Der orangefarbene bzw. dunkelgefärbte Bereich innerhalb des Alphasliders in Abbildung 15 informiert den Anwender noch vor der Betätigung dieses Steuerelementes, dass es keine Filmdatensätze mit „Gerard Depardieu“ als Darsteller und in einem Datum kleiner als etwa 1980 gibt. Betätigt der Anwender den Alphaslider zur Auswahl des Darstellers auf „Spencer Tracy“, bekommt er ohne Zeitverzug die Information, dass für diesen Darsteller keine Filmdatensätze in einem Datum größer als etwa 1970 und kleiner als etwa 1940 existieren. Der Anwender kann dann durch gleicher

⁶ Obwohl es sich bei den entsprechenden Werten nicht um textuelle, sondern numerische Werte handelt, sind hier Alphasliders verwendet worden

Zeit und mit gleichem Aufwand mehr Informationen aufzunehmen, und wird außerdem vor geschützt, eine ungünstige oder gar unsinnige Auswahl zu treffen.

Der Alphaslider stimmt in wesentlichen Punkten mit dem klassischen Schieberegler überein, siehe „5.2 Der Slider (Schieberegler)“. Der Alphaslider benötigt ebenfalls konstanten Platz, unterlaubt ebenfalls weder Mehrfachselektion, noch die Auswahl bestimmter, diskreter Werte (einen entsprechend großen Wertebereich vorausgesetzt).

Die herausragende Eigenschaft des Alphasiders ist die Möglichkeit, nun auch in textuellen Werten zu reichen, zu überbrücken. Zur Auswahl textueller Werte dient unter den klassischen Steuerelementen nur das Texteingabefeld, die Combobox, die Checkbox und der Radiobutton – und mit allen diesen Steuerelementen ist bislang kein Browser möglich.

Daher:

	Platz	Diskrete Werte	Bereich-Auswahl	Mehrfach-Selektion	„browsen“	Visualisierung	Komfort
Alphaslider	+		+		+	+	+

6.2 Das Steuerelement “Data Visualization Slider”

Der „Data Visualization Slider“ ist eine von Stephen G. Eick entwickelte Verbesserung des klassischen Schiebereglers. Das hat vorstechendste Merkmale dieses Steuerelementes, ist es über ein gesamtvier Betriebsmodi verfügt, zwischen denen der Entwickler bzw. der Anwender je nach Bedarf wählen kann:

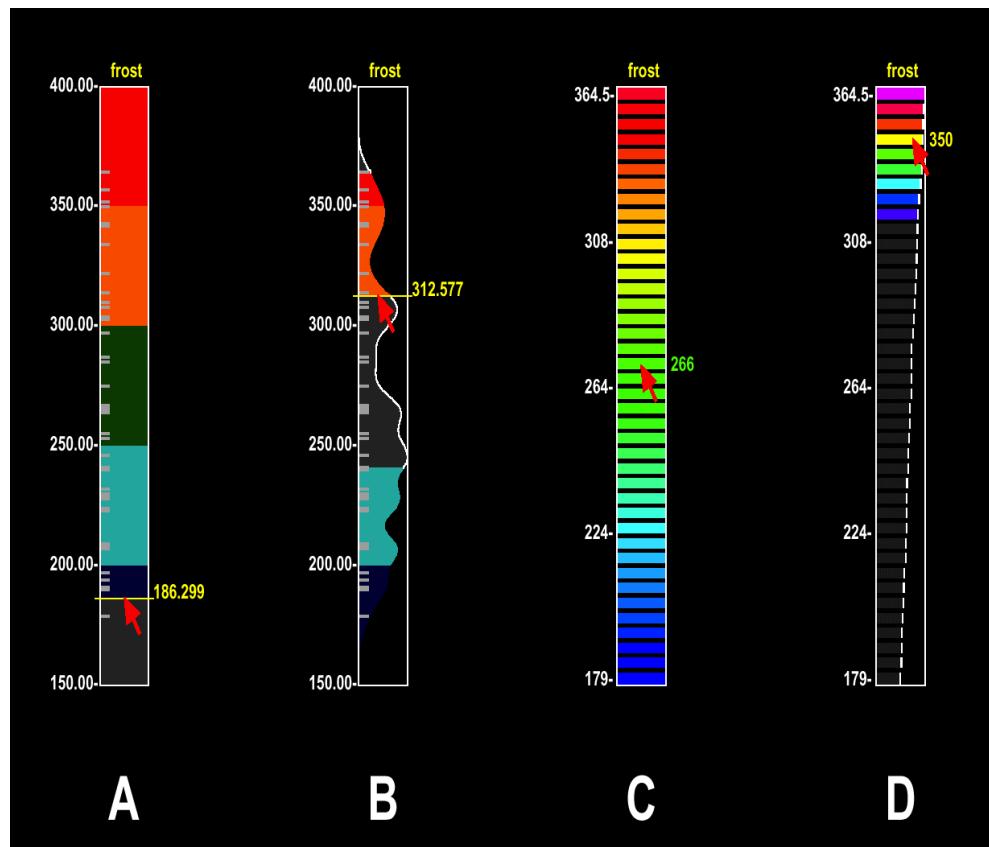


Abbildung 16 – Data Visualization Slider mit 4 Betriebsmodi

Der Data Visualization Slider erlaubt sowohl die Auswahl von Bereichen, als auch die Mehrfachselektion. Zusätzlich dient die Oberfläche des Data Visualization Slider dazu, die Verteilung der zugrunde

liegenden Daten zu visualisieren. Außerdem informiert der Data Visualization Slider den Anwender über den verwendeten Wertebereich.

Der Data Visualization Slider unterscheidet sich optisch sehr stark vom klassischen Schieberegler. Während der klassische Schieberegler aus einem „Knauf“ (andem „gezogen“ wird) in einem „Hintergrund“ (über dem sich der „Knauf“ bewegt) besteht, erscheint der Data Visualization Slider eine einzige, rechteckige, teilweise bunt gefärbte Fläche.

Auch die Vorgehensweise des Anwenders beim Arbeiten mit dem Data Visualization Slider unterscheidet sich etwas von der klassischen Schieberegler. Die Vorgehensweise bei Benutzung des Modus „C“ kann – zerlegt in vier Arbeitsschritte – z.B. wie folgt aussehen:

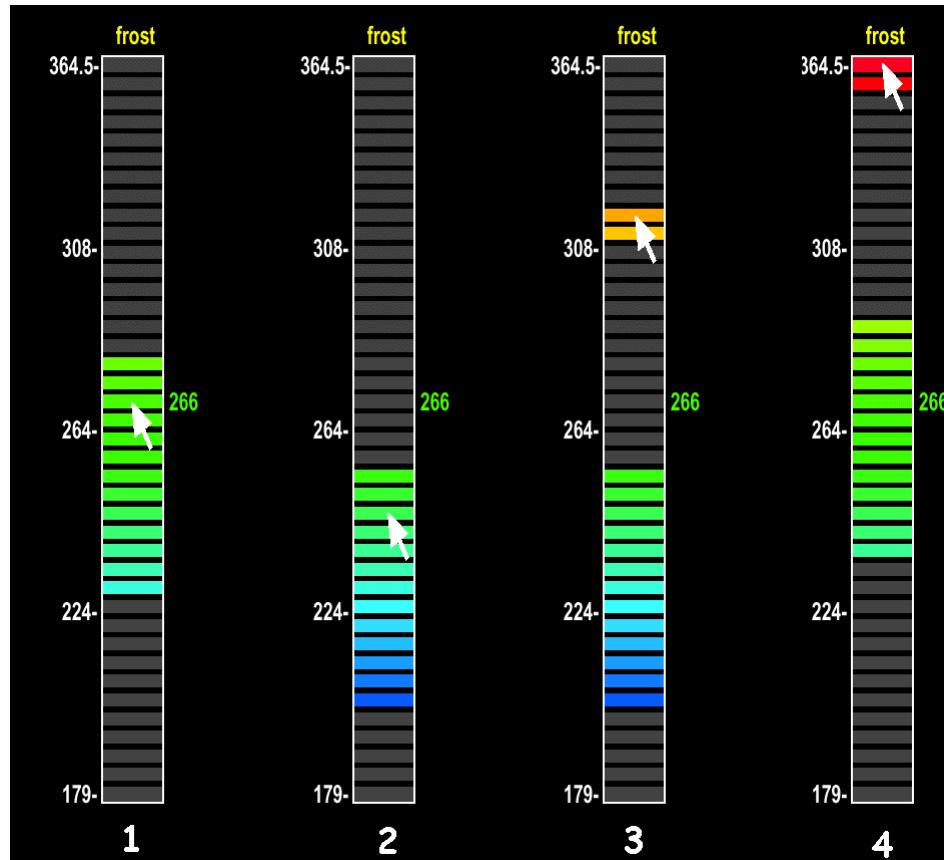
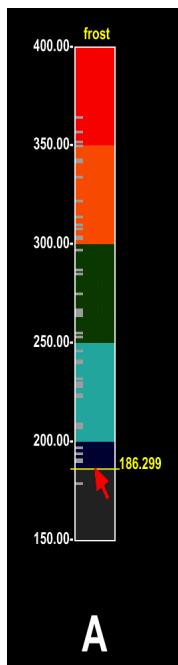


Abbildung 17 -Funktionsweise Data Visualization Slider

1. Zustand 1 ist der Startzustand des Data Visualization Slider. Es sind einerseits dunkelgrau, und andererseits bunt gefärbte Segmente zu erkennen. Die bunt gefärbten Segmente stellen den „Knauf“ des klassischen Schiebereglers dar, während die dunkelgrau gefärbten Segmente den „Hintergrund“ des klassischen Schiebereglers darstellen.
2. Der Anwender verschiebt nun den „Knauf“, indem die Maus darauf bewegt, die Maustaste drückt (ohne sie loszulassen), dann den „Knauf“ an die gewünschte Position bewegt, und die Maustaste wieder loslässt. Das Vorgehen entspricht bisher dem klassischen Schieberegler. Damit ist nun Zustand 2 erreicht.
3. Der Anwender macht nun Gebrauch von der Möglichkeit der Mehrfachselektion – er wählt konkurrenzumübersetzen existierenden Bereich. In einem zweiten Bereich, indem er zu weit ist, klickt er auf den „Knauf“. Der „Knauf“ hat dadurch seine Erscheinung verändert, und damit ist Zustand 3 erreicht.
4. Der Anwender verschiebt nun wieder den „Knauf“, mit dem gleichen Vorgehen wie bereits in Zustand 2.

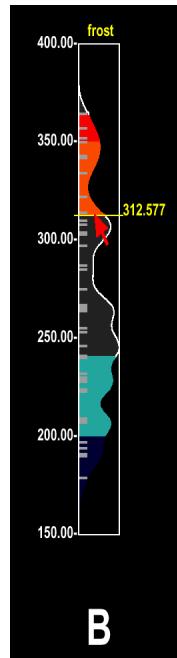
Nachdem nun geklärt ist, wie das Arbeiten mit dem Data Visualization Slider steht, noch die Frage im Raum, worin sich die vier Betriebsarten unterscheiden. Die vier Betriebsarten des Data Visualization Slider unterscheiden sich voneinander darin, wie die Datenverteilung visualisiert wird.



1. Betriebsart A:

In der Betriebsart A wird die Datenverteilung nicht visualisiert. Zwar wird der „Knauf“ eingefärbt, aber die verwendete Farbe steht in keiner Beziehung zur Datenverteilung. Sie dient lediglich der besseren Orientierung während des Arbeitens mit dem Steuerelement.

In der Betriebsart A werden keine diskreten Werte angezeigt. Stattdessen wird dem Anwender suggeriert, dass es mit kontinuierlichen Werten zutun hat – er kann in beliebig kleinen Einheiten⁷ markieren und verschieben.



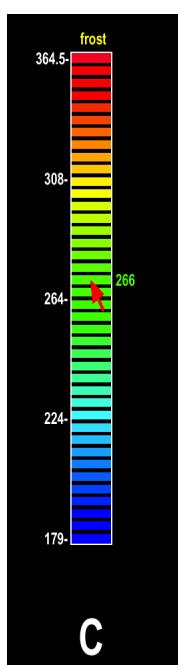
2. Betriebsart B:

Betriebsart B entspricht weitestgehend Betriebsart A, alle Differenzen mit dem feinen Unterschied, dass die Datenverteilung visualisiert wird:

Innerhalb des Data Visualization Slider, d.h. indem Bereich, der eingefärbt wird, wird die Datenverteilung, d.h. die Anzahl der Datensätze, die dieser Schiebereglerposition entsprechen, als Kurve dargestellt.

Abbildung 18 – Data Visualization Slider

Abbildung 19 - Data Visualization Slider

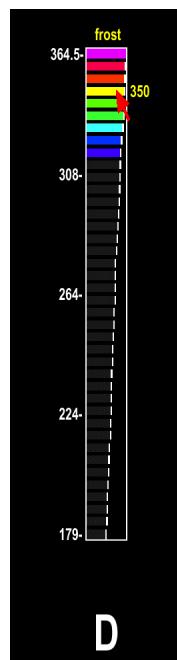


3. Betriebsart C:

Betriebsart C entspricht Betriebsart A mit dem Unterschied, dass der Wertebereich diskret dargestellt wird:

Der Anwender kann lediglich diskrete Werte auswählen, und verschieben.

Für einen „sehr großen“ Wertebereich geht Betriebsart C optisch in Betriebsart A über.



4. Betriebsart D:

Die Unterschiede zwischen Betriebsart C und Betriebsart D entsprechen den Unterschieden zwischen Betriebsart A und Betriebsart B.

Innerhalb des Data Visualization Slider, d.h. indem Bereich, der eingefärbt wird, wird die Datenverteilung, d.h. die Anzahl der Datensätze, die dieser Schiebereglerposition entsprechen, als Kurve dargestellt.

Abbildung 20 – Data Visualization Slider

Abbildung 21 - Data Visualization Slider

⁷ „Beliebig klein“ ist technisch natürlich nicht möglich. Gemeint ist hier, dass zumindest zum mindesten pixelweise selektiert und verschoben werden kann.

Daher:

	Platz	Diskrete Werte	Bereich-Auswahl	Mehrfach-Selektion	„browsen“	Visualisierung	Komfort
DataVisu alizationSl ider	+			+	+	+	+

6.3 Das Steuerelement “2DWidget”

Das 2DWidget ist ein von Ben Schneiderman entwickeltes Steuerelement mit der Funktionalität eines zweidimensionalen Schiebereglers:

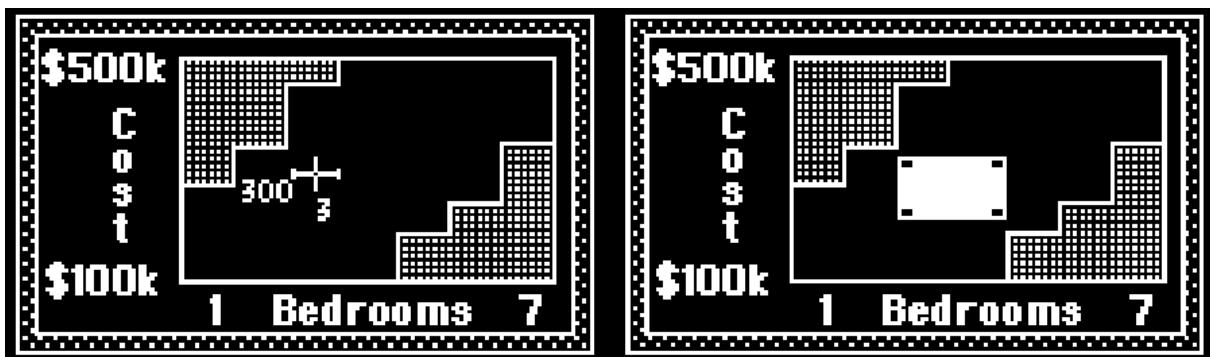


Abbildung 22 -2DWidget

Der Anwender benutzt das 2DWidget, um die Funktion zum Zeichnen eines Rechteckes in einem Zeichenprogramm: Er zieht bei gedrückter Maustaste ein Rechteck auf, und lässt die Maustaste wieder los. Das so gezeichnete Rechteck entspricht dem ausgewählten Wertebereich.

In Abbildung 22 dient ein 2DWidget dazu, eine Eingabe über die Anzahl der Schlafzimmer und den Kaufpreis einer Immobilie zu machen. Der Anwender hat in diesem Beispiel definiert, dass er für eine Immobilie mit zwei Schlafzimmern bis zu 200.000 \$ und für eine Immobilie mit vier Schlafzimmern bis zu 300.000 \$ zu zahlen bereit ist.

Das 2DWidget ist in seiner Lage, sein eigenen Wertebereich zu visualisieren: Dazu werden an den beiden Achsen jeweils zwei Beschriftungen angezeigt. Außerdem existiert innerhalb des Zeichens eine rechteckige, schraffierte Fläche, die nicht ausgewählt werden kann: Indem Beispiel in Abbildung 22 beispielweise beträgt der niedrigste Kaufpreis für eine Immobilie mit sieben Schlafzimmern 310.000 \$ - es ist daher beispielweise nicht möglich, die Kombination „sieben Schlafzimmer“ und „100.000 \$“ zu wählen, da dieser Bereich schraffiert dargestellt ist.

Das 2DWidget bietet neben den bereits genannten Aspekten den Vorteil, dass der Anwender in einer ähnlichen Zeit, in der er mittels eines Schiebereglers eine einzige Größe bestimmt, nüchtern zwei Größen bestimmen kann. Diese Möglichkeiten scheint umso voller, wenn man sie im Zusammenhang mit zweidimensionalen Visualisierungsmethoden wie z.B. dem *Starfield-Display* betrachtet.

Das 2DWidget besitzt einen konstanten Platzbedarf, unterstützt Bereichauswahl und Mehrfachselektion. Weiterhin ermöglicht das 2DWidget das „browsen“, erlaubt aber leider nicht die Auswahl von diskreten Werten.

Daher:

	Platz	Diskrete Werte	Bereich-Auswahl	Mehrfach-Selektion	„browsen“	Visualisierung	Komfort
2DWidget	+		+	+	+		+

6.4 Zusammenfassung

Zusammenfassender ergibt sich für die klassischen neuen Steuerelemente das folgende Bild:

	Platz	Diskrete Werte	Bereich-Auswahl	Mehrfach-Selektion	„browsen“	Visualisierung	Komfort
Slider	+				+		+
Checkbox		+	+	+			+
Radiobutton		+					+
Combobox	+	+					
Alphaslider	+		+		+	+	+
DataVisualizationSlider	+		+	+	+	+	+
2DWidget	+		+	+	+		+

Summa summarum: Die neuen Steuerelemente versprechen neue Möglichkeiten, aber noch immer gilt: Es gibt viel zu tun. Solange das folgende Steuerelement nicht entwickelt worden ist, darf das Ziel dieser Forschung nicht als „erreicht“ bezeichnet werden:

	Platz	Diskrete Werte	Bereich-Auswahl	Mehrfach-Selektion	„browsen“	Visualisierung	Komfort
????????????	+	+	+	+	+	+	+

7 DieLösung(2):MagicLenses

7.1 Einleitung

IndenvorangegangenenKapitelnwurdendieHerausforderungenexpliziert,die *Dynamic Queries* an Steuerelementestellen,unddieProblemeaufgezeigt,beimVersuch,ihnendurchdieErweiterung bestehenderSteuerelementezugegenen.

IndiesemKapitelwirdnuneinalternativerLösungsansatzbeschrieben,dervonKen Fishkinentwickeltwordenist,undaufdenklangvollenNamen,, *MagicLenses* “hört,manchmal auch,, *Movable Filters*“genannt.

HierbeischiebtderAnwender mehreresogenannte,„Linsen“übereinander,dieerselberdefinieren kann,unddiejeweilseinemunkomplizierten,booleaneschenausdruckentsprechen.AufdieseArtund WeisekannderAnwenderspielendbeliebigkomplizierteDatenbankabfragen gestalten.

7.2 LinsenalsFilter

DieIdee das *MagicLenses* istunheimlicheinfach,unddochsehrwirkungsvoll.DieFunktionsweise lässtsichamBestenaneinemBeispielerklären:



Abbildung 23 -MagicLenses -SEL ECT*FROMHOMES

Abbildung 23zeigteneMenge weißenQuadratenaufschwarzemHintergrund,dieseAnordnunglässt sichamBestenals *Starfield-Display*umschreiben.JedesweißeQuadratssymbol isierteinengefundenen Datensatz,dieindiesemBeispielwiedereinzelnenImmobilienentsprechen.

Der Anwender verfeinert nun seine Abfrage „, indem er Stück für Stück weitere Abfragekriterien definiert. Er zieht eine Linse, bindet sie an die Größe „, Kaufpreis“ und setzt ihre Filter „> 225.820“. Diese Linsen schiebt er über den Bereich, Abbildung 23 dargestellten Bereich:

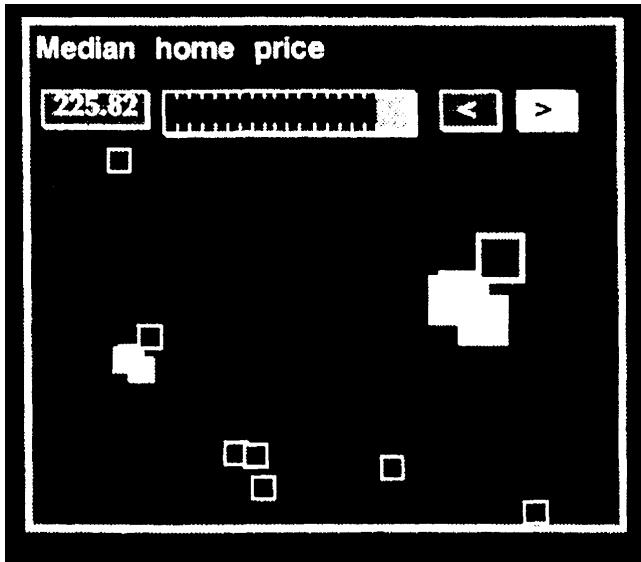


Abbildung 24 - Magic Lenses - Filterfunktion

Die Konsequenz ist, dass alle Immobilien, die diesen Filter nicht „passieren“, durch die Linse nicht mehr als ausgefüllt, sondern als nicht -ausgefüllt, weiße Quadrate dargestellt werden.

7.3 DetailsOnDemand

Findet der Anwender nun Gefallen an einer der Immobilien, und möchte gern eine Info über diese erlangen, so klickt er einfach mit der Maus auf das entsprechende Symbol:

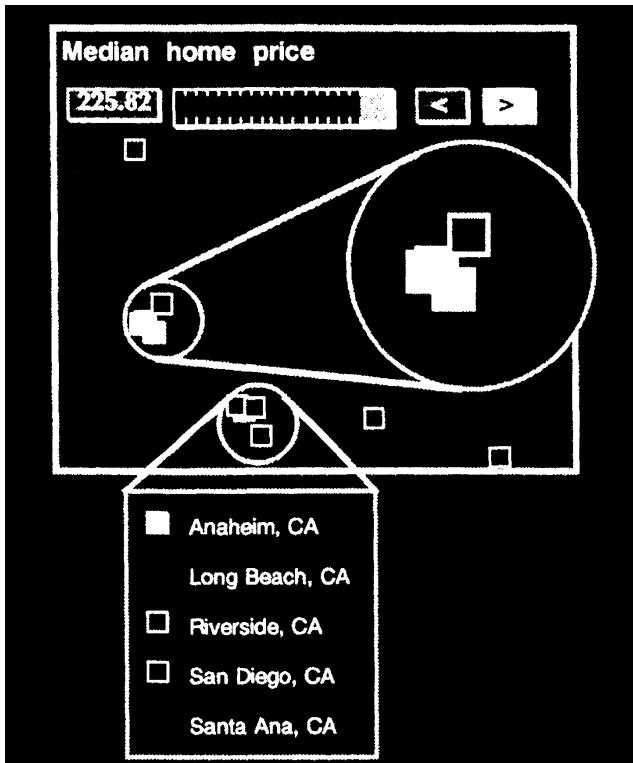


Abbildung 25 - MagicLenses - DetailsOnDemand

Die detaillierten Informationen über die selektierten Symbole werden dann eingeblendet. Diese Technik nennt man auch *Details-On-Demand*.

7.4 Verwendung von mehreren Linsen

Der Anwender kann beliebig viele solcher Linsen definieren, und frei auf dem Bildschirm verschieben; die Darstellung wird dabei in Echtzeit und in die neue Linsenposition angepasst. Schiebt der Anwender zwei (oder mehr) Linsen (partiell) übereinander, so wird die Wirkung der entsprechenden Linsen miteinander verknüpft – ganz so, wie dies auch bei optischen Linsen der Fall wäre:

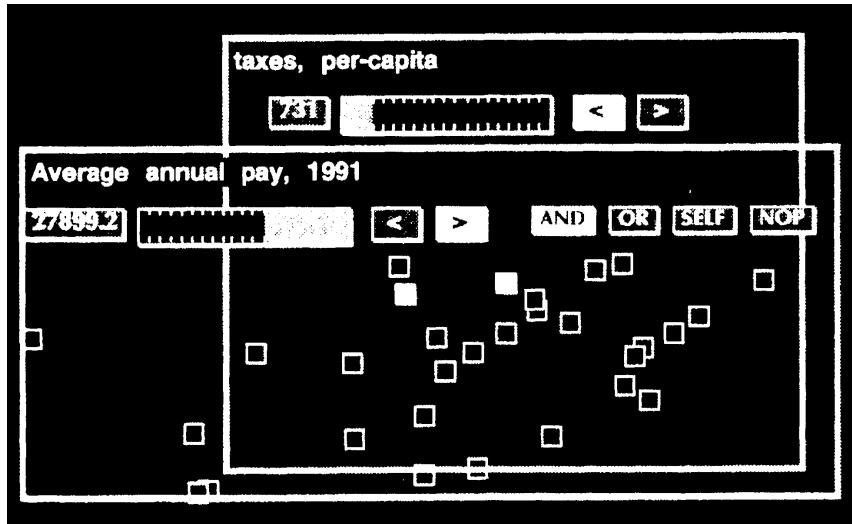


Abbildung 26 -MagicLenses -"und"Verknüpfung

In Abbildung 27 hat der Anwender zwei Linsen übereinander geschoben. Die Folge ist, dass nur noch die Datensätze sichtbar sind, die den Bedingungen beider Linsen genügen. Alle anderen Datensätze werden wieder als nicht-ausgefüllte weiße Quadrate dargestellt.

Der Anwender kann für jede Linse wählen, ob ihr Abfrageergebnis mit dem der darunterliegenden Linsen per „und“-Verknüpfung (AND) oder per „oder“-Verknüpfung (OR) verknüpft wird. Ob die Linse wirkungslos ist (NOP), oder ob sie alle hinterliegenden Linsen wirkungslos machen (SELF).

Betätigt der Anwender nun die Schaltfläche „OR“, so ändert sich die Darstellung ohne Zeitverzug wie folgt:

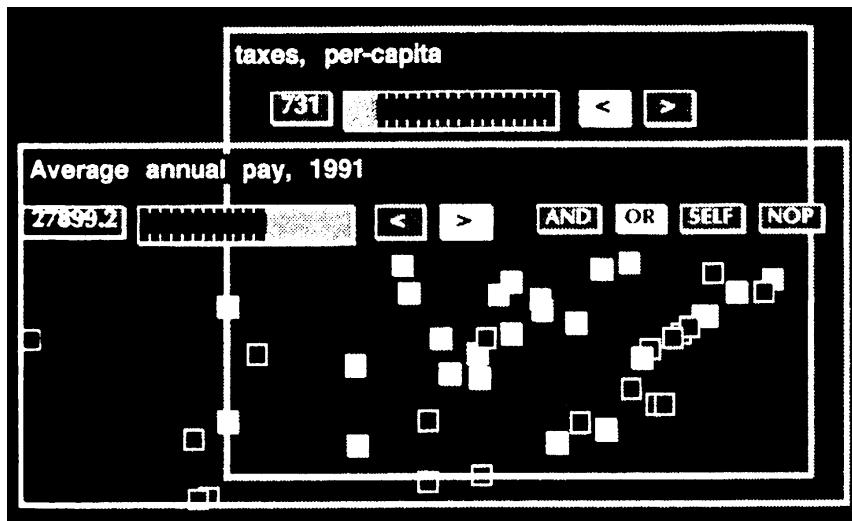


Abbildung 27 -MagicLenses -"oder"Verknüpfung

Mansieht,wiedieAnzahlderausgefülltenQuadrat,e.d.h.dieAnzahldergefundenenDate
merklichangestiegenist.
nsätze,

7.5 Gruppierung

EsbestehtdieMöglichkeit,dieKombinationvonzweidermehrLinsen
sammenzufassen,d.h.zugruppieren.Diese,„Gesamtlinsen“kanndannwieeinzelneLinsebeha
deltwerden,d.h.freiaufdemBildschirm
verschobenwerden.

BezogenaufdieBoolescheAlgebrastelltdieseMöglich
keitdesGruppierensvonLinsen
dieKlamm
erungvonBooleschen(Teil
-)Ausdrückendar.

7.6 FuzzyLogic

EsistderCharakterundzugleichdastypischeProblemerklassischenDatenbankabfrage
,dassder
AnwenderdieundnurdieDatensätze
sieht,nachdenenermittelseinesBooleschenAusdruckesg
e
fragthat.Datensätze,die,„knappdaneben“liegen,werdengenausowenigzurückgegeben,wieDate
n
sätze,die,„totaldaneben“lieg
en.

Die *MagicLenses* bringenhierdieinteressanteMöglichkeitder,, *FuzzyLogic* “insSpiel:Standardm
äßigarbeitenalleLinsen mitUND -und ODER-Operationen,unddenbinären Operanden,,0“ und,,1“.
DerAnwender kanndenModusderLinsenauf,, *FuzzyLogic* “umscha
iten,wasbewirkt,dassfortan
mitMultiplikationundAdditionalsOperationen,undreellenZahlenalsOperandengearbeitetwird.

EinealsFilter einge setzteLinse besitztnunnichtlängerdieRückgabewerte „,0“und,,1“,sondernj
edenreellenWertzwischen,,0“und,,1“ –jenachdem,wiestarkderentsprechendeDatensatz
diedef i
nierteBedingungverfehlt hat.

InderfolgendenAbbildungsindvierDatensätze
inderDetailansichtzusehen,vondenenochnedie
FuzzyLogic nureineeinzelnerDatensatz(„Miami“)dieAbfrag
efilterbiszumEndedurchlaufenhätte.
DurchdieVerwe
ndungder *FuzzyLogic* kannderAnwender
sehen,dasseseinenzweitenDatensatz
„Tampa“gibt,dernurumHaaresbreiten
ebendemAuswahlkriteriumliegt:

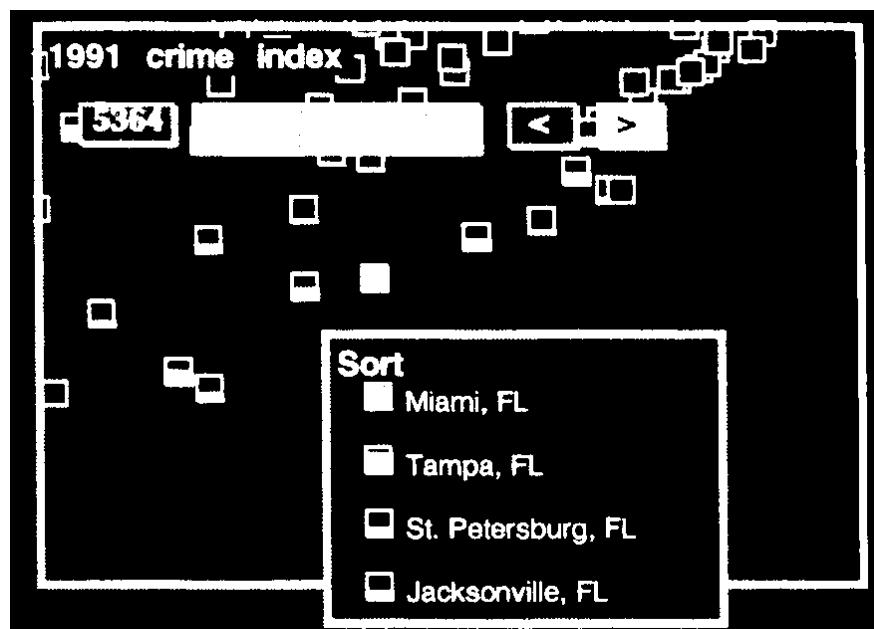


Abbildung 28 -MagicLenses -FuzzyL ogic

7.7 MissingValues

Einweiteres,typischesProblemderklassischenDatenbankabfrage
istdasFehlenvonWerten,das
sogenannte, „MissingValues Problem“ oder, „ MissingData Problem “.WennderDatenb
estandz.B.

auf empirischen Messwerten beruht, kann es gut sein, dass einige Werte fehlen; weil sie sich nicht messen lassen, oder das Messen zu teuer ist.

Bei Verwendung einer klassischen Datenbankabfrage fallen Datensätze mitsolchen, fehlenden Werten zumeist unter den Tisch. Durch die Verwendung einer „MissingData“-Linse räumt die *MagicLens* mit diesem Problem auf:

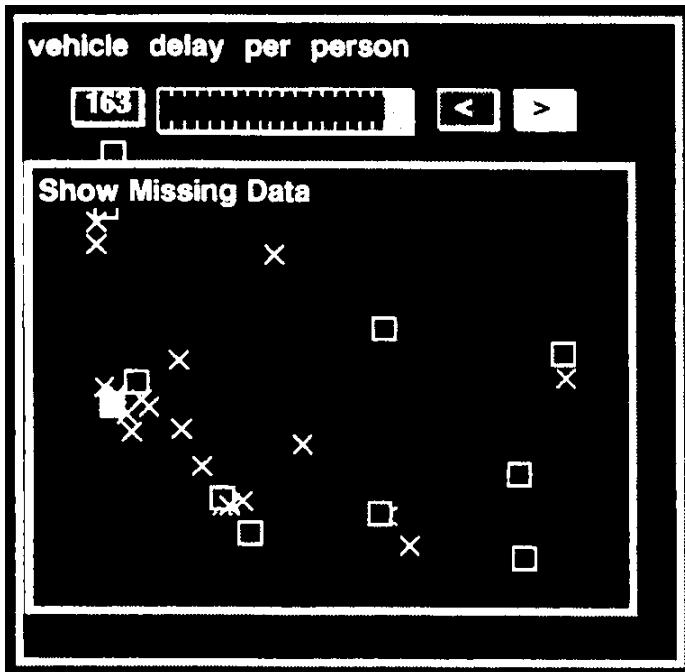


Abbildung 29 -MagicLenses -MissingValues

Der Anwender „schiebt“ die „MissingData“-Linse wie jede andere Linse über die entsprechende Bildschirmposition, und alle Datensätze, die aufgrund eines fehlenden Wertes an einer der dahinterliegenden Linsen gescheitert sind, werden als „X“ dargestellt.

8 Zusammenfassung und Ausblick

DynamicQueries sind mächtig, aber nicht allmächtig. Die neuen, verspielten Möglichkeiten gegeben durch die überden traditionellen Abfragesprachen sind verlockend, aber sie stellen auch neue Herausforderungen, die neuen, denen es zugegessen hat. Es gilt neue Steuerelemente, die Benutzeroberflächen, Datenformate, Algorithmen, Protokolle und Schnittstellen zu entwerfen.

Es ist gezeigt worden, dass auch die neuen Steuerelemente wie *AlphaSlider*, *DataVisualizationSlider* und *2DWidget* nicht in der Lage sind, allen Anforderungen der im Sinne von *DynamicQueries / Tight Coupling* entwickelten Anwendungen zu genügen; und dass es noch zahlreiche Hürden zu überwinden gilt, bis die Visionen der zitierten Vorfreiter in alltägliche Anwendungen software-einzuhalten.

Alle in diesem Papier vorgestellten Anwendungen der *DynamicQueries* sind individuell angefertigte, und auf ein bestimmtes Problem zugeschnittene Speziallösungen. Zukünftig müssen allgemeine, nutzbarbare Werkzeuge um die Fähigkeit der *DynamicQueries* erweitert werden, wie z.B. bestehende Datenbanksysteme und Tabellenkalkulationen – sogenannte „Standardsoftware“. Insbesondere fehlen auch Entwicklungswerkzeuge.

Körperlich behinderte Personen mit Seh- und/oder motorischen Problemen werden durch *DynamicQueries* in der Form, wie sie in diesem Papier präsentiert worden sind, mehr Nachteile als Vorteile erwachsen. Hier gilt es, eine für alle Anwender gleichermaßen nützliche Kompromisslösung zu entwickeln. Denkbar wäre z.B. die Einbeziehung von Audio- und/oder Visualisierung.

Noch immer gilt, dass relativ komplexe Datenbankabfragen wie z.B. „group by“ nur durch die klassischen Datenbankabfrageformulierungen möglich sind. Hier gilt es, noch universellere und vielfältigere Benutzeroberflächen und Steuerelemente zu entwickeln, um mehr und mehr auf die Kommandozeilenzurichten zu können. Als vorläufiges Ziel könnten die hier die Erreichung einer Universalität wie z.B. der von *SQL* stehen.

9 Kommentiertes Literaturverzeichnis

[1] DynamicQueries

(Dynamische Datenbankabfragen)

Card, Stuart K. and Mackinlay, Jock D. and Shneiderman, Ben, *Readings in Information Visualization - Using Visions To Think*, Morgan Kaufmann, 1999, ISBN 1-55860-533-9

Das Sterben nicht - visueller Dienstprogramme ist allgegenwärtig, und macht tauchvorderDate n- bankabfragen ichthalt. Die technologischen Fortschritte in Bezug auf Massenspeicher und Netzwerke haben in einem Overkill an Daten geführt, die für jeden Einzelnenschwer zu bewältigen ist. Neue Abfragetechniken versprechen eine Besserung: Kommandozeilen - Werkzeuge und kryptische SQL - Ausdrücke weichen in der Echtzeit - Simulationen in der virtuellen Realität. Eine Datenbankabfrage wird nicht mehr gestartet, und läuft bis zu ihrem „Ende“, sondern verlässt den Begriff des „Algorithmus“: eine kontinuierliche Interaktion zwischen dem Anwender und dem ausgefeilten Oberfläche vermittelt dem Anwender das Gefühl, eine Abfrage teilzunehmen, oder ein Teil der Abfrage zu sein. Diese neuen Methoden profitieren elegant von den kognitiven Fähigkeiten des menschlichen Auges.

[2] Dynamic Queries for Visual Information Seeking

(Dynamische Datenbankabfragen zur Informationssuche)

Shneiderman, Ben, *Dynamic Queries for Visual Information Seeking*, IEEE Software 11, 6 (1994), 70 - 77.

Dieser Artikel berichtet über Pround Contras von dynamischen Datenbankabfragen gegenüber traditionellen Abfragesprachen wie SQL. Dynamische Datenbankabfragen versprechen kürzere Einarbeitungszeiten, intuitivere und weniger fehleranfällige Möglichkeiten der Abfrageformulierung und eine ansprechende Präsentation des Abfrageergebnisses; sie sind dem Neuling ebenso wie dem Routinier. Dynamische Datenbankabfragen bedeuten aber auch höhere Anforderungen an grafische Ressourcen, neue Herausforderungen für Datenbankstrukturen und Schnittstellen, und Bedarf für die Entwicklung neuer Algorithmen.

[3] Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays

(Visuelle Informationssuche: Die Beziehung zwischen dynamischen Datenfiltern und Starfield-Anzeigen)

4) Ahlberg, Christopher and Shneiderman, Ben, *Visual Information Seeking: Tight coupling of dynamic query filters with starfield displays*, Proc. of ACM CHI'94 Conference (April 1994), 313 - 317 + color plates. Reprinted in Baecker, R.M., Grudin, J., Buxton, W.A. S., and Greenberg, S. (Editors), *Readings in Human-Computer Interaction: Toward the Year 2000*, Second Edition, Morgan Kaufmann Publishers, Inc., San Francisco, CA (1995), 450 - 456.

Dieser Artikel erklärt die als „tight coupling“ (enge Verbindung) bezeichnete Beziehung zwischen der Datenbankabfrage und dem Abfrageergebnis. Die traditionelle Unterscheidung im Sinn von Ursache und Reaktion wird durch neuartige Abfragen über Oberflächen mit Echtzeitverhalten weitestgehend aufgehoben. Steuerelemente greifen der Datenabfrage voraus, indem sie ihren Wertebereich dynamisch anpassen; umgekehrt dient jedes Abfrageergebnis als möglicher Ausgangspunkt für weitere Abfragen. Durch simple Mausklicks werden Abfragen kontinuierlich präzisiert und neu formuliert, der Abfragevorgang wird sich dadurch in einem Frage - Antwort-Dialog ineinander „browsen“. Hierbei spielen engen Datenanzeigen und Steuerelemente eine besondere Rolle; die Autoren heben dabei vor allem die „Starfield - Anzeige“ und den „Alpha slider“ hervor, und demonstrieren deren Funktionsweise anhand der selbst entwickelten Applikationen „Film Finder“ und „Dynamic Home Finder“.

[4] DataVisualizationSlider s

ScrollbalkenzurDatenvizualisierung

S. G. Eick, AT&T Bell Laboratories, Quelle#1, Seiten 251 - 252)

DerklassischeSchieberegler ist seit U rbeginne in fester Bestandteile der grafischen Benutzeroberflächen. S. G. Eick stellt in diesem Artikel eine Weiterentwicklung dieses Steuerelementes vor, speziell getrimmt auf den Einsatz in dynamischen Datenbanken. Es fragt. Ganz im Sinne von „Tight Coupling“ wird das Ergebnis der Datenbank abfragedirekt innerhalb des Schiebereglers präsentiert. Trotzgleichem Platzbedarf bietet er weiterte Schieberegler Möglichkeiten wie Bereichsauswahl und Mehrfachselektion.

[5] EnhancedDynamicQueriesviaMovableFilters

Erweiterte Datenabfrage mit beweglichen Filtern

K. Fishkin und M. C. Stone, Xerox PARC, Quelle#1, Seiten 253 - 259)

Dieser Artikel beschreibt den Spagatzwischenmobersten Gebot, „intuitive Bedienung“ und dem Wunsch, damit alle Datenbankabfragen formuliert werden können. Die Autoren stellen eine Linsenlösung dieser Problematik vor, die führen in die Idee der „beweglichen Filter“ oder auch „magischen Linsen“ ein. Beide sind Prinzipien der Datenbankabfragedefinition, die Anfragersteller auf der Oberfläche „Linsen“, die in einem booleschen Ausdruck entsprechen. Diese Linsen können beliebig über einander geschoben, neu definiert, gruppiert und wieder gelöscht werden. Trotzdem ist eine einfache und verspielte Benutzerführung auf diese Art und Weise jede Datenbankabfrage realisierbar.

[6] DynamicHomeFinderProjectPlanSection 1: Introduction and Overview

„Dynamic HomeFinder“ Projektplanung, Teil 1: Einführung und Übersicht

Christopher Williamson und Tom Smallwood, 1995

Christopher Williamson und Tom Smallwood, die Autoren der erweiterten Version von „Dynamic HomeFinder“ erklären hier die Ziele und Grundlagen ihres Softwareprojektes, und führen in die Software ein.

Dieses Dokument wurde unter der URL <http://www.dqsoft.com/homefind/dhintro.htm> gefunden.

10 Abbildungsverzeichnis

Abbildung1 -Dynamic HomeFinder -SELECT*FROMHOMES	5
Abbildung2 -Dynamic HomeFinderwährend einer typischen Abfrage	6
Abbildung6 -Ein Schieberegler	10
Abbildung7 -Einige Kontrollkästchen	11
Abbildung8 -Ein Opti onsfeld	12
Abbildung9 -Ein Kombinationsfeld	12
Abbildung10 -Alphaslider	14
Abbildung11 -Alphaslider (Bereichsmarkierung)	14
Abbildung12 -Alphaslider Kontextmenü 1	15
Abbildung13 -Alphaslider Kontextmenü 2	15
Abbildung14 -Alphaslider Visualisierung 1	15
Abbildung15 -Alphaslider Visualisierung 2	15
Abbildung16 -DataVisualizationSlider mit 4 Betriebsmodi	16
Abbildung17 -Funktionsweise DataVisualizationSlider	17
Abbildung18 -DataVisualizationSlider	18
Abbildung19 -DataVisualizationSlider	18
Abbildung20 -DataVisualizationSlider	18
Abbildung21 -DataVisualizationSlider	18
Abbildung22 -2DWidget	19
Abbildung23 -MagicLenses -SELECT*FROMHOMES	21
Abbildung24 -MagicLenses -Filterfunktion	22
Abbildung25 -MagicLenses -DetailsOnDemand	22
Abbildung26 -MagicLenses -"und" Verknüpfung	23
Abbildung27 -MagicLenses -"oder" Verknüpfung	23
Abbildung28 -MagicLenses -FuzzyLogic	24
Abbildung29 -MagicLenses -MissingValues	25

11 Index

2

2DWidget·19,20,26

A

Abfrage·4,6,22,27

Abfrageergebnis·4,6,23,27

Aktion·4,8

Algorithmus·4,26,27

Alphaslider·14,15,16,20, 26,27

Anwender·4,5,6,7,8,9,10,11,12,13,15,
16,17,18,19,21,22,23,24,25,26,27

Anwendung·4,5,6,7,8,15

Ausnahme·12

B

Benutzerführung·28

Benutzeroberfläche·6,10,26,28

Bereichsauswahl·9,10,11,12,28

Bildschirm·9,10 ,11,23,24

Binär·24

C

CDE·10

Checkbox·10,11,12,13,16,20

Combobox·10,12,13,16,20

D

DataVisualizationSlider·16,17,18,19,20,
26,28

Datenbankabfrage·4,5,6,21,24,25,26,27,
28

Datenbankserver·5

Datensatz·6,9,18,21,23, 24,25

DetailsOnDemand·22,23

Dokument·28

F

Filter·21,22,24,28

Funktion·19

FuzzyLogic·24

I

Interaktion·4,27

K

KDE·7,10

Kommandozeile·26

L

Linse·21,22,23,24,25,28

Liste·9,12,13

M

MagicLenses·21,22,23,24,25

Massenspeicher·4,27

Maus·6,10,17,22

Mehrfachselektion·9,10,11,12,16,17,19,
28

Menge·6,9,21

Microsoft·7,10

MissingData·24,25

MissingValues·24,25

N

Netzwerk·4,27

O

Oberfläche·4,6,7,16,27,28

OpenLook·10

P

PalmOS·10

Protokoll·4,26

R

RadioButton·12,13,16,20

Ressource·4,9,27

Rückgabewert·24

S

Schaltfläche·8,10,23
Schieberegler·7,8,9,10,11,12,13,14,16,
17,18,19,20,28
Schnittstelle·4,26,27
Software·5,15,27,28
SQL·4,5,26,27
StarfieldDisplay·6,19,21,27
Steuerelement·1,4,5,6,7,8,9,10,13,14,
16,18,19,20,21,26,27
Swing·7,10
Symbol·6,22

T

Texteingabefeld·8,16

U

URL·5,28

W

Wertebereich·4,8,9,10,11,14,15,16,17,
18,19,27
Windows·7,10