

Dynamic Queries

Proseminar Informationsvisualisierung SS 2000

Johann-Wolfgang-Goethe Universität Frankfurt – Fachbereich Informatik

Martin Klossek – martin@klossek3000.de

Inhaltsverzeichnis

1	EINLEITUNG	3
1.1	ÜBERSICHT.....	3
2	DYNAMIC QUERIES	4
2.1	TRADITIONELLE ABFRAGETECHNIKEN	4
2.2	TRADITIONELLE DARSTELLUNGSFORMEN.....	4
2.3	VISUELLES ABFRAGEN.....	5
2.4	FILMFINDER.....	7
2.5	MENSCHLICHE REAKTIONSZEITEN	9
3	STARFIELD DISPLAYS	11
3.1	AUSGABE VON DYNAMIC QUERIES.....	11
3.2	NATÜRLICHE MAPPINGS	11
3.3	KONSTRUIERTE MAPPINGS	13
4	TIGHT COUPLING	14
4.1	KOPPLUNG VON EINGABE UND AUSGABE	14
4.2	OPTIMIEREN VON SOFTWAREOBERFLÄCHEN	15
5	SCHLUSSWORT	18
6	ANHANG	19
6.1	QUELLENVERZEICHNIS	19
6.2	INDEX.....	21

1 Einleitung

Durch den in den letzten Jahren und Jahrzehnten erheblich angestiegenen Einsatz immer leistungsfähigerer Informationstechnik und Rechnersysteme sind die zu verwaltenden Datenmengen im gleichen Atemzug rapide angewachsen. Durch die intensiviertere Datenerfassung in allen Bereichen der Wirtschaft, Industrie und Forschung aber auch im privaten Bereich werden immer größere Ansammlungen von Daten produziert.

Jüngstes Beispiel hierfür sind die Bemühungen um die Sequenzierung des menschlichen Erbguts wie sie im Humane Genome Project oder von privaten Unternehmen betrieben werden¹.

Gewaltige Datensammlungen alleine helfen dem Menschen aber nicht, seinen Wissenshunger zu stillen, neue Erkenntnisse zu gewinnen und Zusammenhänge zu erkennen. Gemäß dem Übergang von Daten zu Informationen und schließlich zu Wissen, müssen Daten aufbereitet werden, um dem Menschen von Nutzen zu sein. Mit der Flut von Bits und Bytes sind aber selbst geübte Anwender von Rechnersystemen überfordert, wenn sie in Form von endlosen Zahlenkolonnen vorliegen und Anfragen an die Technik nicht ergonomisch sind.

Abhilfe schaffen alternative Benutzerinterfaces, die die menschlichen Bedürfnisse und physischen wie psychischen Besonderheiten unseres Wesens besser zu berücksichtigen versuchen. Ein Ansatz hierzu sind Visualisierungen von Datenbeständen. Andere Wege sind durch den Einsatz von Multimedia und die Berücksichtigung weitere Sinne als Auge und Ohr denkbar.

So ist diese Ausarbeitung zum Thema Dynamic Queries in ihrem Grundsatz eine Betrachtung alternativer, optimierter Benutzerinterfaces, die zwar keinen umfassenden Überblick über Visualisierung geben wird, aber sich mit dem Teilbereich der grafischen Repräsentation von Daten mit einem Schwerpunkt in der Benutzerinteraktion beschäftigt.

1.1 Übersicht

Die Ausarbeitung gliedert sich in die drei Teile Dynamic Queries, Starfield Displays und Tight Coupling mit einer Vorstellung der einzelnen Oberbegriffe. Dabei wird unter Dynamic Queries eine Gegenüberstellung traditioneller Datenabfrage- und Ausgabetechniken und der neue Ansatz der grafischen Auswahl beschrieben. Mit Starfield Displays werden Konzepte zur Visualisierung von Datenbeständen auf zweidimensionalen Koordinatensystemen vorgestellt und im letzten der drei Teile wird unter dem Begriff Tight Coupling die Verknüpfung von grafischer Ein- und Ausgabe erläutert.

¹ Informationen zu diesem Projekt siehe beispielsweise unter <http://www.dhgp.de/>

2 Dynamic Queries

2.1 Traditionelle Abfragetechniken

Größere Datenbestände werden in unterschiedlicher Form in Datenbanken gespeichert. Je nach Ausprägung des Systems können Anfragen in Form von programmierter Logik erfolgen, indem beispielsweise direkt auf Dateien von Massenspeichern zugegriffen wird oder abstrakter über vom Datenbanksystem zur Verfügung gestellte Abfragesprachen wie SQL oder XQL.

Hier soll es unerheblich sein, wie die Daten letztlich gespeichert sind und mit welcher konkreten Methode auf sie zugegriffen wird. Gemeinsam ist diesen traditionellen Abfragearten aber, dass in allen Fällen vom Anwender Befehle zusammengesetzt werden müssen. Hierbei müssen insbesondere die gewünschten Datenquellen gewählt und eine Auswahl der auszugebenden Werte vorgenommen werden.

Wurde ein so konstruierter Befehl an das Datenbanksystem gesendet, erhält der Anwender in der Regel eine tabellarische Rückgabe. Abbildung 1 illustriert diesen Vorgang mit einem einfachen SQL-Befehl, der alle Datensätze, bei denen das Attribut „feld“ gleich „xyz“ ist, aus einer Tabelle anfordert.

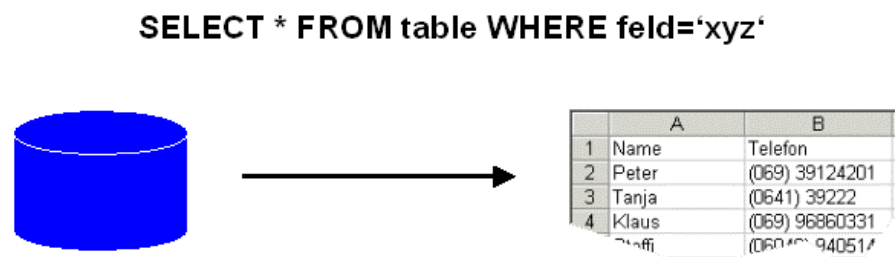


Abbildung 1 - Datenbankabfrage mit SQL

Zwar sind mit solchen Datenabfragetechniken sehr individuelle und leistungsfähige Anfragen möglich, der Aufwand zur Konstruktion entsprechender Befehle oder gar festverdrahteter Programmlogik ist gleichzeitig aber erheblich. Die Wahrscheinlichkeit, eine aufgrund zu starker Einschränkungen des Wertebereichs leere oder falsche Ausgabe zu erhalten ist hoch. Auch leichte Modifikationen der Eingabeparameter mit dem Ziel, der gesuchten Ausgabe näherzukommen, gestalten sich hier schwierig.

Zudem muss eine genaue Detailkenntnis der Implementierung vorliegen, um die gewünschten Datenquellen und Beschränkungen des Wertebereichs richtig zu wählen. Gerade bei großen Datenbeständen kommt es so auch schnell zu unübersichtlichen Ausgaben, da die Einschränkungen zu weit gewählt waren.

Eine alternative Möglichkeit, seitens des Anwenders Datenabfragen zu spezifizieren, wäre daher sinnvoll. Dynamic Queries bieten diese Alternative. Zunächst aber eine Betrachtung der traditionellen Darstellungsformen von Daten.

2.2 Traditionelle Darstellungsformen

Traditionelle Computersysteme geben ihre Datenbestände häufig in der Form aus, wie sie im Rechner gespeichert sind und richten sich somit eher nach den Bedürfnissen der Technik als denen des Anwenders. So liefern die im vorangegangenen Abschnitt erläuterten Abfragetechniken ihre Ausgaben in tabellarischer Form zurück.

Die visuelle Ausgabe auf Bildschirm oder als Ausdruck erfolgt entsprechend häufig in Tabellenform. Dabei werden die einzelnen zurückgelieferten Datensätze Zeile um Zeile gemäß der gewünschten Sortierung ausgegeben. Ebenso oft finden sich auch Formulare

oder Masken, die das Anzeigen und Bearbeiten von einzelnen Datensätzen erlauben. Eine weitere Repräsentationsform sind Berichte, die allerdings schon eine Interpretation und Gruppierung von Datensätzen vornehmen und nicht mehr die reinen Daten zurückliefern.

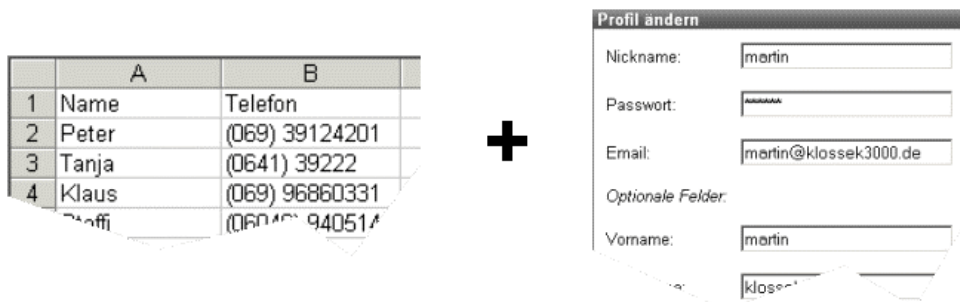


Abbildung 2 - Datenausgabe in Tabelle und Formular

Vorteil diese Darstellungsformen von Daten ist ihre programmtechnisch einfache Implementierung und die universelle Einsetzbarkeit für jede Form von gespeicherten Daten. Übersichtlich können insbesondere Tabellenausgaben bei sehr vielen Datensätzen nicht sein. Es werden zu viele Datensätze bei zu vielen Details angezeigt. Eine Reduktion der angezeigten Datenmenge und ein geschickteres Layout versprechen hier Abhilfe.

Ein umfassender Überblick über den Datenbestand und die Möglichkeit Zusammenhänge erkennen zu können, sind hier nur sehr schwer zu erzielen. Dynamic Queries mit geeigneten Visualisierungen können die Probleme der tabellarischen Datenausgabe für einige Anwendungsbereiche lösen.

2.3 Visuelles Abfragen

Mit Dynamic Queries lassen sich die erwähnten Probleme der textuellen Datenabfrage und -ausgabe wie komplizierte Eingabe der Abfragebefehle oder unübersichtliche Ausgaben lösen. Die Handlungen des Anwenders verschieben sich vom fehlerträchtigen Suchen zum spielerischen Browsen im Datenbestand.

Bei Benutzeroberflächen, die auf den Ideen von Dynamic Queries aufbauen, steht dem Anwender eine Arbeitsfläche zur Verfügung, die einerseits Steuerelemente zum Spezifizieren der Suchanfrage und andererseits einen Ausgabebereich für die darzustellenden Informationen enthält. Der Ausgabebereich kann aber auch gleichzeitig Eingabeaufgaben erfüllen, wenn beispielsweise ausgegebene Elemente wie Linien, Punkte oder Piktogramme vom Benutzer angewählt werden können und eine Veränderung der dargestellten Werte bewirken. Genauso kann die Ausgabe in die Steuerelemente zurückfließen und zu einer sinnvollen Einschränkung des wählbaren Wertebereichs führen. Diese Verbindung von Ein- und Ausgabe wird mit dem Begriff Tight Coupling beschrieben, der weiter unten aufgegriffen wird.

Die Neuartigkeit des Visuellen Abfragens – im folgenden gleichbedeutend mit Dynamic Queries verwendet – hat ein Verschwinden der Grenzen von Eingabe- und Ausgabeverhalten von Benutzeroberflächen zur Folge. Vorteilhaft daran ist, dass es hier keine Trennung mehr zwischen dem Spezifizieren der Abfragebefehle und dem Betrachten der Ausgabe gibt. So wären auch traditionelle Systeme denkbar, die auf Datenbankabfragen keine Tabellen sondern Diagramme als Ausgabe liefern. Visuelles Abfragen geht einen Schritt weiter, da auch das Anfragen visuell erfolgt und Datenbankinternas dem Anwender verborgen bleiben.

Ermöglicht wird das visuelle Zusammenstellen von Suchanfragen durch von grafischen Oberflächen² bereits angebotene Steuerelemente wie Buttons, Auswahllisten oder

² Oberflächen wie sie bei Microsoft Windows, X, KDE, MacOS oder OS/2 verwendet werden

Schieberegler sowie neuartige Eingabelemente wie Alphaslider, Data Visualization Slider oder Magic Lenses. Alle diese Steuerelemente dienen dazu, Einschränkungen der Datenausgabe vorzunehmen. Beispielsweise kann wie beim FilmFinder, einer im nächsten Abschnitt vorgestellten Beispielapplikation, mit Hilfe eines Schiebereglers die Menge der dargestellten Filme auf die gewünschte Jahreszahl von einem Start- zu einem Endjahr festgelegt werden. Die Ausgabe der Filme in einem Koordinatensystem repräsentiert durch Punkte erfolgt unmittelbar beim Bewegen des Reglers. Die Reaktionszeiten der Software sollten sehr schnell sein, so dass die technischen Anforderungen an Geschwindigkeit und Speicherplatz von Rechnersystemen bei Dynamic Queries recht hoch werden können.

Ein typischer Arbeitsablauf zum Auffinden von Daten bei einer traditionellen Datenbankanwendung wäre das Formulieren eines Abfragebefehls in SQL, XQL oder einer anderen Abfragesprache oder alternativ das Ausfüllen eines Eingabeformulars mit den gewünschten Suchbegriffen. Nach Abschicken der Suchanfrage an das System würden kurze Zeit später die entsprechenden Daten zurückgeliefert und in Tabellen, Formularen, Diagrammen oder einer anderen Repräsentationsform ausgegeben. Je nachdem, wie gut die Anfrage gestellt war, erhält der Anwender die gewünschten Informationen. Häufig entspricht die Ausgabe aber nicht dem gewünschten Suchziel, so dass die Anfrage mit modifizierten Parametern von Neuem gestartet wird. Der Zeitaufwand zwischen Spezifizieren der Suchanfrage und Begutachten der Ausgabe sowie der generelle Aufwand zum Erreichen des Suchziels kann erheblich sein, da kontinuierlich zwischen den beiden Programmteilen Suche und Ausgabe gewechselt wird.

Beim Visuellen Abfragen ist diese Trennung wie schon weiter oben angesprochen aufgehoben. Ein typischer Arbeitsablauf zum Suchen von Information sähe hier beispielsweise so aus, dass dem Anwender zunächst eine grobe aber umfassende Übersicht des Datenbestandes in einer geeigneten Visualisierung gegeben wird. Dabei ist es nötig, auf die Ausgabe von einigen Informationen zu verzichten und somit eine Reduktion der ausgegebenen Daten vorzunehmen. Beispielsweise werden für jeden Datensatz unter Verzicht der Darstellung von Attributen wie Beschreibung oder Größe nur Punkte ausgegeben.

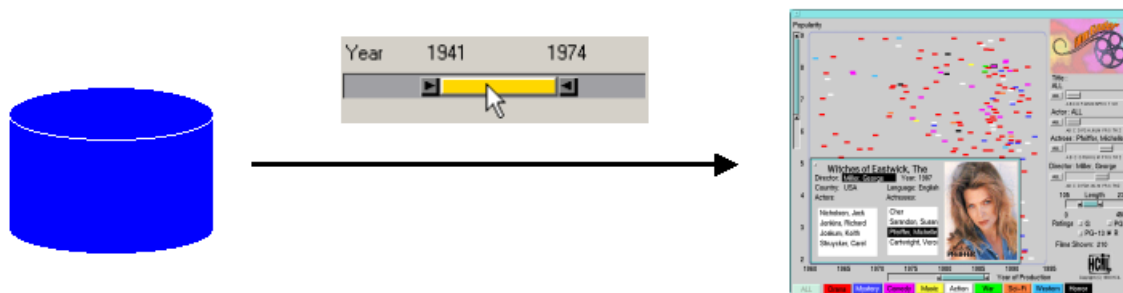


Abbildung 3 - Datenbankabfrage mit Dynamic Queries

Mit Hilfe von Steuerelementen wie Buttons, Auswahllisten und Schieberegler ist der Anwender in der Lage, diese grobe Übersicht zu verfeinern, indem er in den Datenbestand hereinzoomt³, die Menge der dargestellten Werte einschränkt durch die Wahl von Grenzwerten der jeweiligen Attribute eines Datensatzes oder detaillierte Informationen zu einem Wert anfordert (Details-on-demand). Bei diesem Verfeinerungsvorgang werden jeweils die Datensätze ausgegeben, die innerhalb der mit den Steuerelementen gewählten Grenzen liegen. Die restlichen Elemente des Datenbestandes werden dabei ausgeblendet.

Genauso ist das Erweitern einer eingeschränkten Ansicht möglich, indem die Steuerelemente in die der Einschränkung jeweils umgekehrte Richtung bewegt werden. Sowohl beim Verfeinern als auch beim Erweitern der Anfrage erfolgt die Ausgabe unmittelbar

³ hier ist das Vergrößern und Verkleinern des dargestellten Datenausschnitts gemeint

– vorzugsweise ohne nennenswerte Verzögerungszeiten. Das Arbeiten mit Dynamic Queries Systemen gestaltet sich somit wie ein Browsen, dass man auch von Programmen zum Anzeigen von Webseiten kennt, bei denen dem Anwender eine vorgegebene Menge von Verzweigungen zu anderen Informationseinheiten angeboten wird⁴, die ein schnelles Auffinden von Daten ermöglichen und die dahinterstehende Technik verbergen.

Daraus resultieren Vorteile visueller gegenüber textueller Abfragen von Datenbanksystemen wie das spielerische Erzielen von Abfrageergebnissen ohne hohen Einarbeitungsaufwand oder die Möglichkeit, Zusammenhänge anhand der vorgenommenen Visualisierung und der schnellen Anpassungsfähigkeit der angezeigten Werte zu erkennen. Die direkte Veränderung der Suchanfrage zwischen fest vorgegebenen Grenzwerten verhindert zudem die Wahrscheinlichkeit, leere oder ungewollte Ausgaben zu erhalten. Aufgrund der flexiblen Regler und der vorzugsweise schnellen Antwortzeit des Systems können ungewünschte Ansichten schnell in die gesuchten Ausgaben überführt werden. Im folgenden Abschnitt werden die hier geschilderten Ideen hinter Dynamic Queries anhand einer exemplarischen Applikation – dem FilmFinder – illustriert.

2.4 FilmFinder

Anfang der neunziger Jahre entwickelten Wissenschaftler und Studenten um Professor Ben Shneiderman, der an der „University of Maryland“, USA tätig ist, Konzepte zum Visuellen Abfragen. Dabei entstand auch eine Demonstration dieser Abfragetechnik: Der FilmFinder, ein Werkzeug zur Recherche in einer Filmdatenbank.

Hier ist zudem der damalige Student Christopher Ahlberg zu nennen, der an der Entwicklung dieser Anwendung beteiligt war und später mit einer eigenen Firma eine kommerzielle Version dieses Produktes entwickelt hat, das sich auch für andere Anwendungen beispielsweise aus der Betriebswirtschaft verwenden lässt⁵.

Ziel dieser Software ist es, einen großen Bestand an Filminformationen – ungefähr 1500 Filme sind enthalten – mit einem einfach zu bedienenden Benutzerinterface einzusehen. Denkbar wäre beispielsweise das Szenario einer vierköpfigen Familie, die einen gemeinsamen Fernsehabend plant. Gemäß den Präferenzen der einzelnen Familienmitglieder wird eine Auswahl der Filmgenres getroffen, die minimale und maximale Spieldauer und möglicherweise ein bevorzugter Schauspieler gewählt. Mit dem FilmFinder ist das problemlos möglich. Die gewünschten Werte werden einfach mit den Steuerelementen eingegrenzt und sofort erfolgt die Ausgabe der Filme, die den gewählten Kriterien entsprechen, mit jeweils kleinen Rechtecken, deren Farbe ihrem Genre entspricht, in einem Koordinatensystem. Auf der horizontalen Achse ist das Produktionsjahr und auf der vertikalen der Grad der Popularität aufgetragen. Die Ausgabe der Filme – das Mapping auf ein Koordinatensystem – erfolgt also in einem Punktgraph, der im weiteren auch als Starfield Display bezeichnet wird. Abbildung 4 zeigt die Software in Aktion.

Im Beispielszenario der den Fernsehabend planenden Familie würde dann eine Ausgabe der möglichen Filme erfolgen, die den gewählten Vorlieben der einzelnen Familienmitgliedern entsprechen. Bestände die gefundene Menge beispielsweise aus zehn Filmen, könnten mit der Software zu jedem Film weitere Informationen angefordert werden. Denkbar wäre beispielsweise auch, eine solche Software als Online-Dienst eines Video-on-demand⁶-Anbieters zu installieren und nach Wahl eines Filmes diesen dann an den Besteller auszuliefern.

⁴ Verzweigungen innerhalb von Hypertextdokumenten durch Hyperlinks oder Auswahlmenüs

⁵ die von Christopher Ahlberg gegründete Firma unterhält eine Website unter <http://www.spotfire.com>

⁶ Technik, die Videodaten via „Kabel“ oder Telefonleitung an Haushalte ausliefert, die sich aber bislang aus unterschiedlichen Gründen nicht durchsetzen konnte

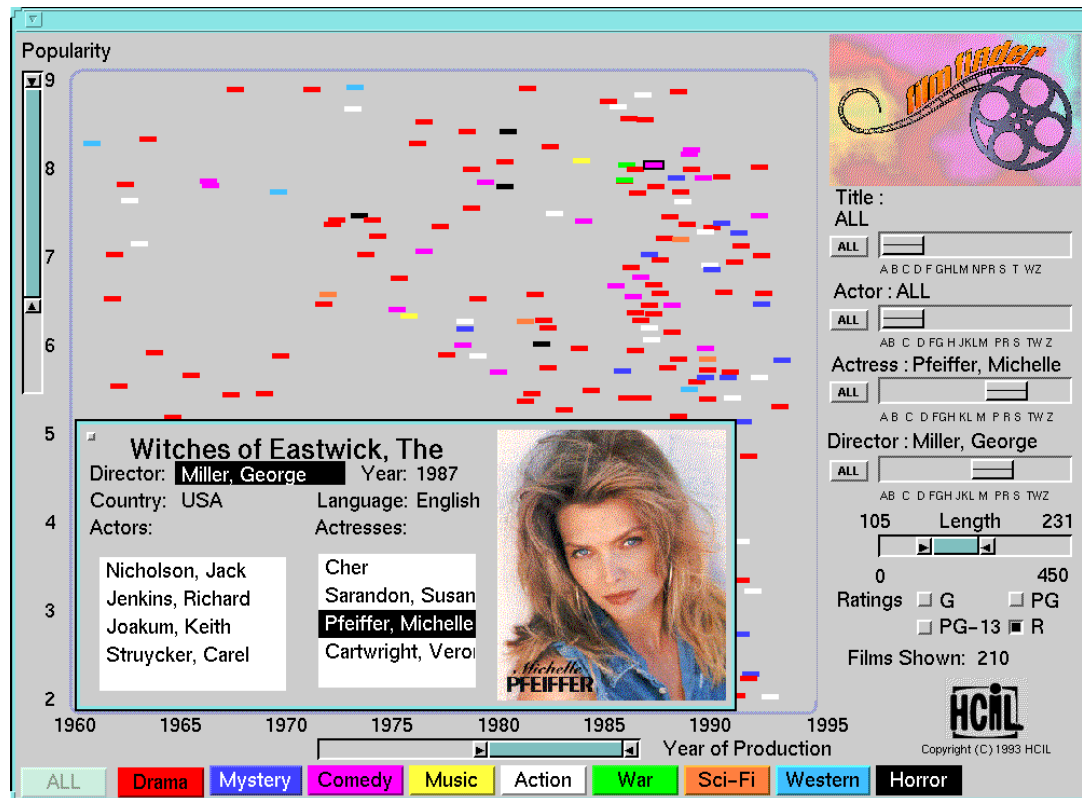


Abbildung 4 - Screenshot des FilmFinders in Aktion

Der Bildschirmaufbau des FilmFinder besteht im wesentlichen aus zwei Teilen, einem Ausgabebereich (Starfield Display), der hier sehr deutlich durch die hohe Anzahl an farbigen Rechtecken zu erkennen ist und einem Bereich, der die Steuerelemente beinhaltet. Jedes der Rechtecke im Starfield Display repräsentiert einen Film. Die Farbe entspricht dem jeweiligen Genre. Die gleichen Farbe werden auch für die Genreschalter im unteren Bereich des Fensters verwendet. In der Ausgabe werden nur die Filme angezeigt, für die der entsprechende Genreschalter aktiviert ist. Damit liegt eine Oder-Verknüpfung des Attributs Filmgenre für die Datenabfrage vor.

Mit den Steuerelementen im rechten Bereich kann man Eingrenzungen des Wertebereichs verschiedener Attribute wie Filmlänge (zweiseitiger Schieberegler) oder Popularität (Checkboxes) vornehmen. Neuartig sind dabei die Alphasliders für den Titel des Films, die beteiligten Schauspieler und Schauspielerinnen sowie Regisseure. Mit diesem Steuerelement können alphanumerische Werte über einen Schieberegler ausgewählt werden. Im Starfield Display werden dann nur die Filme angezeigt, bei denen der entsprechende Titel, Schauspieler und Regisseur (jeweils mit Unschärfe) gefunden wird. Zwischen diesen Elementen besteht die für Dynamic Queries typische Und-Verknüpfung.

Der große Rahmen mit dem Foto stellt eine mit Details-on-demand⁷ beschriebene Technik dar. Jedes der einen Film repräsentierenden Rechtecke kann mit einem Zeigergerät (typischerweise eine Computermaus) angeklickt werden, worauf sich ein Fenster mit Detailinformationen zum entsprechenden Film öffnet. Dieses Verfahren bietet die Möglichkeit, mehr Informationen zu einem Datensatz (hier: ein Film) anzuzeigen, als es in dem als Übersicht genutzten Starfield Display möglich ist. Aufgrund des begrenzt zur Verfügung stehenden Anzeigeplatzes werden nur kleine Rechtecke angezeigt. Sinkt die Anzahl der dargestellten Rechtecke auf ein kleines Maß ab, werden zu jedem Rechteck noch die Filmtitel ausgegeben. Die Reduktion der Ausgabe auf das Mindestnötige mit dem Ziel eine Übersicht zu bieten, unterschlägt zunächst detaillierte Information zu den einzelnen Datensätzen, die mit Details-on-demand wieder angeboten werden.

⁷ kann direkt übersetzt als „Details-auf-Abruf“ genannt werden

Interessant ist beim FilmFinder auch, dass eine Verknüpfung von Eingabe und Ausgabe – das sogenannte Tight Coupling – erfolgt. Einerseits kann die Ausgabe in Form des Starfield Displays als Eingabeelement genutzt werden, wenn der Anwender die Rechtecke anklickt. Andererseits wird der Auswahlbereich eines Steuerelemente auf der rechten Seite in Abhängigkeit von der Einstellung der anderen Steuerelemente eingeschränkt oder erweitert. Hier fließt dann die Ausgabe zurück in die Eingabeelemente. Vorteil ist, dass der Anwender so mit hoher Wahrscheinlichkeit eine Einstellung der Regler wählt, die ein sinnvolles Ergebnis produziert. Andernfalls wären Konstellationen der eingestellten Werte der Steuerelemente denkbar, die sich gegenseitig ausschließen und so keine Ausgabe von Filmen im Starfield Display erzeugen.

Das auffälligste Merkmal an der FilmFinder-Software ist das neuartige Gefühl bei der Bedienung der Software, die ein wenig an Computerspiele erinnert. Bewegungen an den Alphaslidern, das Aktivieren und Deaktivieren von Genrebuttons oder die Begrenzung und Erweiterung der Spiellänge führt zu einer unmittelbaren Veränderung des Punktgraphen. Rechtecke verschiedener Farbe erscheinen und verschwinden wieder. Auch das Zoomen in den Auswahlbereich hinein und heraus mit den Reglern an den Achsen des Koordinatensystems oder das Verschieben dieses Sichtfensters laufen ohne merkliche zeitliche Verzögerung ab. Durch Anklicken eines Rechtecks erhält man zudem weitere Informationen. So werden Suchergebnisse durch spielerisches Browsen, was nicht abwertend zu sehen ist, erzielt und ohne das komplizierte Verfassen von Abfragebefehlen oder Ausfüllen von Formularen.

2.5 Menschliche Reaktionszeiten

Für das unmittelbare Verändern der Ausgabe bei Dynamic Queries sind schnelle Algorithmen und schnelle Hardware nötig. Traditionelle Datenbanksysteme haben häufig zu langsame Antwortzeiten, um die schnell veränderliche Datenauswahl zu visualisieren. Für Dynamic Queries sind daher je nach Struktur des Datenmaterials angepasste Datenstrukturen nötig, um schnelle Veränderungen an den Einstellungen der Steuerelemente unmittelbar auf die Ausgabe zu übertragen. Wie hoch oder niedrig müssen aber diese Reaktionszeiten sein?

Eine Einteilung menschlicher Reaktionszeit ist in drei Vorgänge möglich. Der erste Vorgang ist die Zeitspanne, die zwei Ereignisse zeitlich mindestens voneinander entfernt sein müssen, damit sie vom Menschen wahrgenommen werden. Man spricht vom „Psychologischen Moment“, der ungefähr 0,1 Sekunden bzw. 100 Millisekunden dauert. Zwei Ereignisse können beispielsweise zwei Töne oder zwei visuelle Signale sein.

Psychologischer Moment (psychological moment)	0,1 Sekunden
Spontane Reaktion (unprepared response)	1 Sekunde
Kognitiver, interaktiver Job (unit task)	5 - 15 Sekunden

Tabelle - Einteilung menschlicher Reaktionszeiten

Der zweite Vorgang beschreibt die Reaktionszeit des Menschen auf unvorhergesehene Ereignisse. Beispielsweise dauert das Erkennen von plötzlich auftretenden Hindernissen im Straßenverkehr ca. eine Sekunde. So erkennt man den spontan einsetzenden Bremsvorgang des vor dem eigenen fahrenden Fahrzeugs mit einer knappen Sekunde Verzögerung. Die Zeitspanne wird „Spontane Reaktion“ genannt.

Der dritte und längste Vorgang schließlich beschreibt die Zeit, die ein Anwender benötigt, um Einheiten von Operationen auszuführen. Gemeint ist dabei nicht das Schreiben eines Textes als intellektuelle Aufgabe, sondern beispielsweise das Neuformatieren eines bereits geschriebenen Textes. Für den im Englischen mit „unit task“ bezeichneten Vorgang wird eine Zeitspanne von fünf bis 15 Sekunden angesetzt.

Besonders die erste Zeit ist für Computersoftware relevant, die schnell auf Interaktion mit dem Benutzer ausgelegt sind. Bei Visuellen Abfragen muss die Veränderung der Datenausgabe, die durch den Benutzer mit Bedienung der Steuerelemente ausgelöst wird, innerhalb des Psychologischen Moments erfolgen, damit die Aktion „Bedienen des Steuerelements“ und „Veränderung der Datenausgabe“ als ein einziger Vorgang erscheint und so dem natürlichen Empfinden entspricht, dass beispielsweise die Bewegung eines Türgriffs die damit verbundene Tür öffnet.

Um diese hohe Reaktionsgeschwindigkeit zu erreichen, ist eine Anpassung der Software an menschliche Reaktionszeiten nötig, die durch schnelle Hardware und gute, an die Problemstellung angepasste Algorithmen und Datenstrukturen möglich wird. Dadurch stellt man Echtzeitverhalten von Software ein, die die Konzepte des Visuellen Abfragens umsetzt.

3 Starfield Displays

3.1 Ausgabe von Dynamic Queries

Nachdem die Auswahl der anzuzeigenden Wertemenge mit Hilfe von Steuerelementen erfolgt und die Daten geliefert wurden, müssen sie in einer an die Thematik und Problemstellung angepassten Form ausgegeben werden. Insbesondere ist eine schnelle Ausgabe wichtig, um dem Benutzer das Verändern von Eingabeelementen und das Anzeigen von Werten als einen Schritt erscheinen zu lassen. Schnell zu implementierenden Visualisierungsverfahren ist daher prinzipiell der Vorzug bei Dynamic Queries zu geben.

Hierzu zählen beispielsweise Punktgraphen - auch Starfield Displays genannt -, die von Grafikhardware schnell berechnet und ausgegeben werden können, die mit Hilfe von unterschiedlicher Farbcodierung der Punkte dennoch aussagekräftig sind und zudem durch geringe Platzbeanspruchung der einzelnen Werte ein übersichtliches Bild des dargestellten Ausschnitts vermitteln. Hier besteht auch die Möglichkeit, da die dargestellte Information auf einen Farbpunkt reduziert ist, diese per Mausklick zu selektieren und so weitere Informationen anzuzeigen (Details-on-demand).

Denkbar sind auch andere Repräsentationsformen der Daten wie verschiedene Diagrammtypen (Balken, Torten, Linien), sich dynamisch verändernde Tabellen oder auch dreidimensionale Visualisierungen. Wichtiges Augenmerk ist aber immer eine möglichst hohe Geschwindigkeit, da es sich hier nicht um statische Grafikausgabe sondern letztlich um Grafiken handelt, die durch die Interaktion des Benutzers bewegt werden müssen. In allen Fällen spricht man beim Übertragen der Daten auf eine Darstellungsform von Mapping.

Die Wahl des geeigneten Mappings hängt entscheidend vom Anwendungsbereich und der Art der darzustellenden Daten ab. Für Applikationen, die geografische Daten darstellen, bieten sich Graphen an, die die Daten auf eine Karte in Form der betrachteten Region zeichnen. Für andere Anwendungsfälle sind solche „natürlichen Mappings“ nicht möglich, hier können abstrakte Achsenbeschriftungen für Koordinatensysteme gewählt werden.

3.2 Natürliche Mappings

Im Falle von Anwendungen, die auf natürlichen Karten basieren, können die Daten direkt hierauf abgebildet werden. Der Assoziationseffekt beim Betrachter ist dadurch sehr hoch, da vertraute visuelle Gebilde wie Karten von geografischen Regionen mit Punkten oder anderen eingezeichneten Objekten eine neue Bewertung erhalten. Allerdings muss es sich nicht notwendigerweise um geografische Karten handeln. Auch andere, dem Anwender vertraute visuelle Gebilde können als Basis für Dynamic Queries und als Starfield Displays dienen, wenn die Problemstellung entsprechende Grafiken bereithält. Im folgenden werden einige Beispiele für solche „natürlichen Mappings“ gegeben.

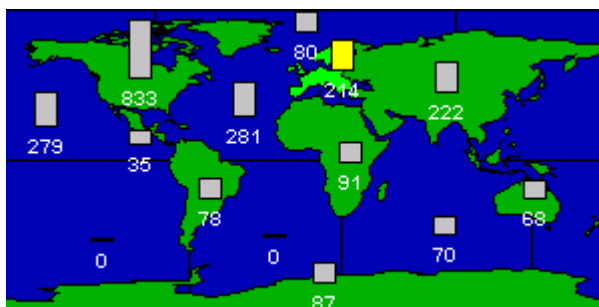


Abbildung 5 - Mapping von Werten auf Weltkarte

Für Anwendungsbereiche, in denen geografische, klimatische oder wirtschaftliche Daten in Bezug auf Regionen, Kontinente oder gar die ganze Welt dargestellt werden sollen, ist ein

Mapping der Daten auf das Bild einer entsprechenden Karte als Hintergrund denkbar. Die vom Anwender per visueller Abfrage gelieferten Werte können in Form von Punkten, Kreisen oder auch Rechtecken dargestellt werden. Erfolgt die Ausgabe der Daten in Abhängigkeit von einem Attribut der Datensätze, beispielsweise der Größe eines Wertes, so kann der Wert dieses Attributes mit einer Skalierung des verwendeten Symbols ausgegeben werden. Dadurch wäre ein Vergleich zwischen einzelnen Regionen der Karte realisierbar.

Ein Beispiel für eine Europakarte mit Kreisdiagrammen in jedem Land, die die Gewichtungen bestimmter am rechten Rand im Kontrollbereich gewählter Größen symbolisieren.

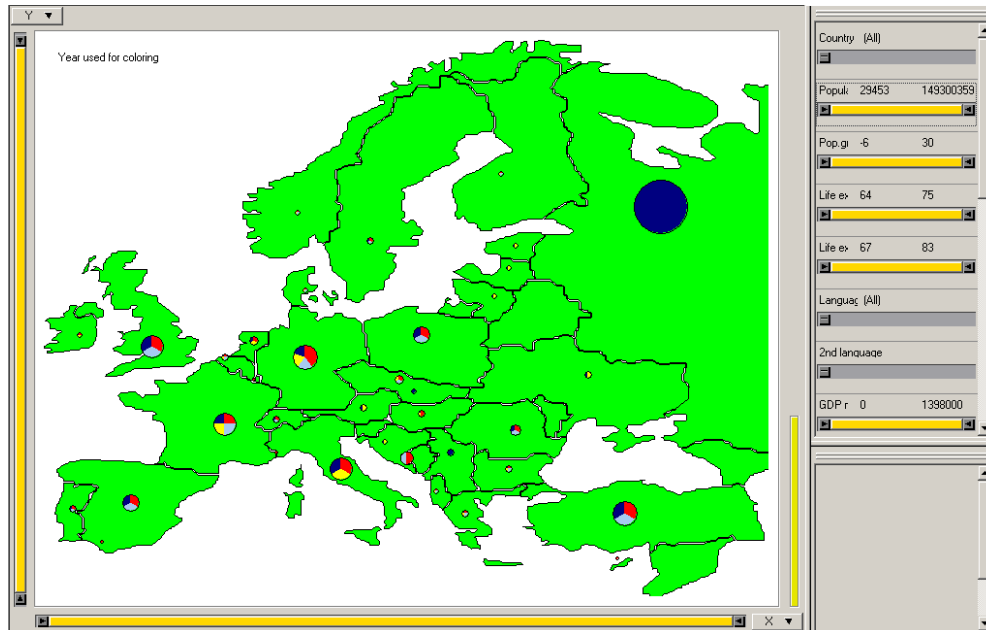


Abbildung 6 - Geografische Daten zu Europa (aus der Software Spotfire Pro)

„Natürliche Mappings“ sind wie eingangs angesprochen nicht nur auf Problemstellungen anwendbar, die auf Kartenmaterial basieren, sondern auch auf andere Bereiche, die mit etablierten Grafiken, Tabellen oder Diagrammen arbeiten. Ein Beispiel hierfür sei das in der Chemie verwendete „Periodensystem der Elemente“.

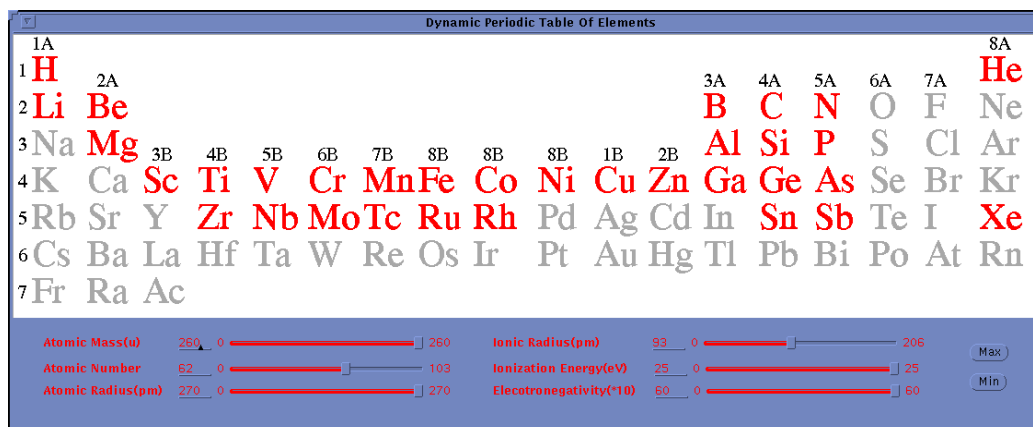


Abbildung 7 – Mapping auf das Periodensystem der Elemente (Chemie)

Im Periodensystem werden Atome nach verschiedenen Eigenschaften geordnet, beispielsweise nach Anzahl der freien Elektronen⁸ bzw. fehlenden Elektronen (in Bezug auf die Edelgaskonfiguration) in den Spalten. In einer Applikation könnten durch Bewegen von

⁸ Elektronen sind negativ geladene Teilchen oder auch Wellen von Atomen

Schieberegler für „Atomgewicht“ oder „Anzahl der freien Elektronen“ die Bezeichner der Atome rot gemalt werden, für die die gewählte Klassifikation zutrifft oder grau, wenn sie nicht zutrifft.

3.3 Konstruierte Mappings

In den vorangegangenen Beispielen standen Grafiken, Karten oder andere Bilder bereit, auf denen Daten visualisiert werden können und so eine „natürliche“ Abbildung der Daten auf ein Koordinatensystem gegeben ist. Ist ein solches „natürliche Mapping“ nicht möglich, können die Daten in ein abstraktes zweidimensionales Koordinatensystem aufgetragen werden. Für die beiden Achsen werden der Problemstellung angepasste Beschriftungen gewählt. Denkbar wäre auch, die Wahl der Achsenvariablen in der Software veränderbar zu machen, so dass verschiedene Aspekte und Attribute der Datensätze betrachtet werden können.

Anstelle zweidimensionaler Koordinatensysteme sind auch andere Darstellungsformen vorstellbar. Beispielsweise Datenausgabe in Kreisdiagrammen oder dreidimensionale Darstellungen. Wichtig ist auch hier, dass das Visualisieren der Daten mit hoher Geschwindigkeit erfolgen muss, um dem Anwender schnelle Änderungen der Ausgabe zu ermöglichen.

In dieser Ausarbeitung sollen nur die zweidimensionalen Ausgaben betrachtet werden, da der Schwerpunkt des Themas nicht auf der Visualisierung der Daten sondern auf der Interaktion von Visualisierungssystemen mit dem Benutzer liegt. Punktgraphen oder Starfield Displays eignen sich hervorragend als Ausgabetypp, da sie so implementiert werden können, dass sie schnelle Aktualisierungen bieten und so die verzögerungsfreie Interaktion mit dem Benutzer gestatten.

Ein solches Starfield Display wurde beim FilmFinder, der eingangs vorgestellten Beispielapplikation für Dynamic Queries, eingesetzt. Auf der horizontalen Achse ist das Produktionsjahr und auf der vertikalen Achse die Spiellänge aufgetragen. Alle im Koordinatensystem eingezeichneten Punkte repräsentieren Filme. Weiter links liegende Filmpunkte haben ein älteres Produktionsdatum als weiter rechts liegende und Filmpunkte, die weiter oben aufgetragen werden, haben eine längere Spieldauer als solche, die weiter unten zu finden ist. Mit Schieberegler besteht in der Implementierung die Möglichkeit, sowohl die Spanne der Produktionsjahre als der Spieldauer zu verkleinern und vergrößern sowie zu verschieben. Entsprechend werden nur die Filme als Punkte ausgegeben, die in den gewählten Bereich reinfallen. Auch können andere Variablen als Produktionsjahr und Spieldauer für die Achsen gewählt werden und so eine andere Sicht auf den Datenbestand erfolgen.

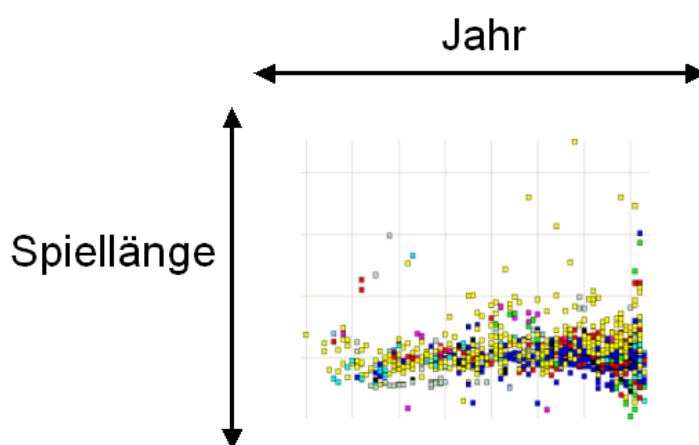


Abbildung 8 - Wahl eines geeigneten Koordinatensystems (für FilmFinder)

4 Tight Coupling

4.1 Kopplung von Eingabe und Ausgabe

Visuelle Abfragetechniken ersetzen klassische Datenbankabfragen und Visualisierungsformen wie Starfield Displays werden als Alternative zu nüchternen, tabellarischen Zahlenkolonnen eingesetzt. Die vormalig bestehende Grenze zwischen dem Formulieren von Abfragen und dem anschließenden Betrachten der Ausgabe wird bei Dynamic Queries zu einem Browsen mit kontinuierlichem Wechselspiel zwischen visuellem Suchen und Auswerten.

Allerdings können auch bei Dynamic Queries Arbeitsabläufe noch kantig sein. Es besteht beispielsweise die Gefahr, dass Steuerelemente in solche Einstellungskonstellationen gebracht werden, die bei einer erfolgten Anfrage keine entsprechenden Datensätze liefern und der Anwender so eine leere Ausgabe erhält. Hier setzt ein mit Tight Coupling⁹ beschriebenes Softwaredesign an, das die Ausgabe von Datenbankabfragen nicht nur in die dafür vorgesehene Zeichenfläche sondern auch zurück in die Steuerelemente leitet, mit denen die Abfrage formuliert wurde. Ebenfalls wird die Ausgabe mit einem Anwendungsdesign, das Tight Coupling verfolgt, selbst zum Eingabeelement, wenn beispielsweise gezeichnete Elemente auf der Zeichenfläche angewählt werden und dann eine Reaktion der Anwendung auslösen (zum Beispiel für Details-on-demand).

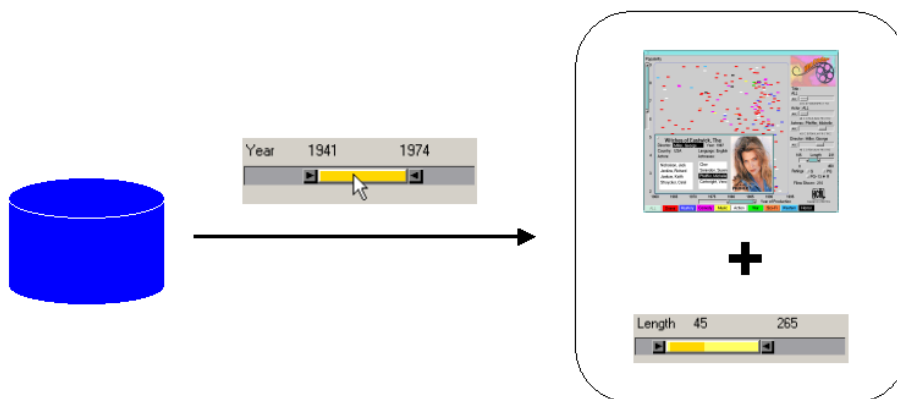


Abbildung 9 – Tight Coupling lenkt Ausgabe in Starfield Displays und Eingabeelemente

Ein weitere Idee, die zu einer bedienungsfreundlicheren Programmoberfläche führt, ist es, dem Benutzer eines Dynamic Queries System eine Reihe von Startabfragen zur Auswahl zu geben, die durch vorgegebene Steuerelementeinstellungen eine Ausgabe erzeugen, mit der der Anwender seinen Browservorgang starten kann. Solche Starteinstellungen können zum Beispiel eine Gesamtübersicht des Datenbestandes oder markante Datenausschnitte sein, deren nähere Betrachtung lohnt. Vorteil an solchen Presets ist, dass der Anwender bei erstmaliger Benutzung der Software Ergebnisse ohne größeren Einarbeitungsaufwand erzielen kann und so sein Suchziel schneller zu erreichen in der Lage ist.

Durch die Rückkopplung von Ausgabe und Steuerelementen, wird die Ausgabe der Daten gleichzeitig zur Eingabe. Der Benutzer kann dann im günstigsten Fall nur sinnvolle Abfragen realisieren. Dadurch wird gleichzeitig eine intuitive Benutzerführung realisiert, da im aktuellen Programmzustand nicht ausführbare Befehle nicht anwählbar sind oder erst gar nicht dargestellt werden. Das Beispiel des Textverarbeitungsprogramms mit dem aktivierten und deaktivierten Speichern-Menüeintrag verdeutlicht das Prinzip des Tight Coupling, das zu einem Vermeiden von überflüssigen oder gar falschen Eingaben dient.

⁹ übersetzt bedeutet Tight Coupling so viel wie „Enge Kopplung“

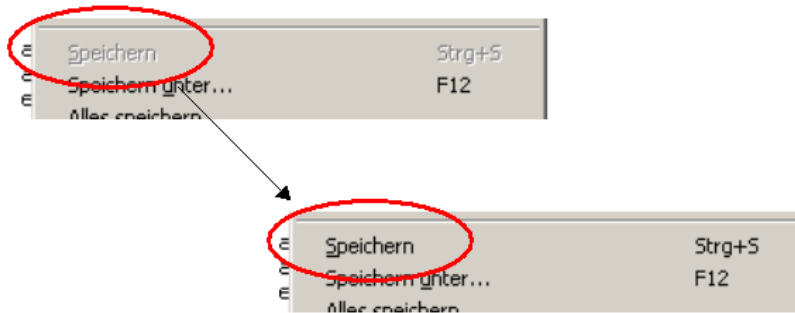


Abbildung 10 - Aktivierung des Speichern-Menüeintrags

Startet der Anwender sein Textverarbeitungsprogramm und öffnet ein neues Dokument, so ist die Schreibfläche zunächst leer und der Speichern-Menüeintrag deaktiviert. Der Benutzer kann sein Dokument also nicht speichern, da es zum gegenwärtigen Zeitpunkt keine sinnvolle Aktion wäre, das Dokument ist schließlich leer.

Gibt der Anwender Text und andere Bestandteile seines Dokuments in die Software ein, wäre ein Speichern sinnvoll. Der Speichern-Menüeintrag wird daher vom Programm aktiviert und der Benutzer kann sein Dokument speichern. Nach dem Speichervorgang ist der Menüeintrag wieder deaktiviert, da ein nochmaliges Speichern keine Veränderung des Dokumentes ergeben würde. Erst sobald eine Veränderung am Inhalt vorgenommen wird, beispielsweise die Korrektur eines Wortes oder das Hinzufügen eines neuen Absatzes, wird das Speichern wieder möglich.

Diese einfache Beispiel für das Tight Coupling Prinzip zeigt, wie Bedienungsoberflächen von Software verbessert werden können und so ein fehlerärmeres und damit zügigeres Arbeiten möglich wird. Im folgenden Abschnitt wird noch einmal auf das Optimieren von Softwareoberflächen mit Tight Coupling bei Dynamic Queries eingegangen.

4.2 Optimieren von Softwareoberflächen

Wie schon im vorangegangenen Abschnitt beschrieben, wären Dynamic Queries Systeme ohne Tight Coupling kantig zu bedienen und würden deutlich mehr „leere Ausgaben“ und Frustration des Anwenders erzeugen. Allerdings handelt es sich bei Tight Coupling weniger um einen fest umrissenen Begriff als vielmehr um das Prinzip für ein Softwaredesign mit möglichst bedienungsfreundlicher Oberfläche. Daher können verschiedene Aspekte einer Dynamic Queries Software unter Tight Coupling aufgezählt werden.

So sollten Steuerelemente immer als solche erkennbar sein und im gesamten Programmaufbau durchgängig eingesetzt werden, beispielsweise immer gleiches Layout bei Buttons, Auswahllisten oder Schiebereglern. So wäre es eher nachteilig für den Bedienungskomfort des Anwenders, wenn Buttons einmal mit abgerundeten und einmal mit kantigen Ecken innerhalb verschiedener Programmbereiche verwendet würden. Auch ist die verständliche Beschriftung der Steuerelemente ein wichtiger Faktor, um den Anwender nicht unnötig zu verwirren.

Wichtig für Dynamic Queries Systeme ist auch die Reversibilität von Aktionen. So müssen Steuerelemente einerseits schnelle Ausgabeveränderungen sowohl in kleinen als auch großen Schritten zulassen, andererseits aber auch in der Lage sein, diese in der nächsten Aktion zurückzunehmen. Dadurch wird dem Anwender ein Browsen oder Spielen mit der Software möglich, da Fehlbedienung nicht zu Datenverlust führt und jede Operation durch Reversibilität sofort zurückgenommen werden kann.

Wie beim Verfeinern von SQL-Abfragen durch Erweitern oder Einschränken der Wertegrenzen, lässt sich mit Tight Coupling die Ausgabe verfeinern. Allerdings geschieht dies hier wesentlich intuitiver und weniger fehleranfällig, da kleine Veränderungen an den

Steuerelementen nur kleine Veränderungen an der Ausgabe bewirken und die Resultate in Form veränderter Ausgabe in Sekundenbruchteilen sichtbar werden.

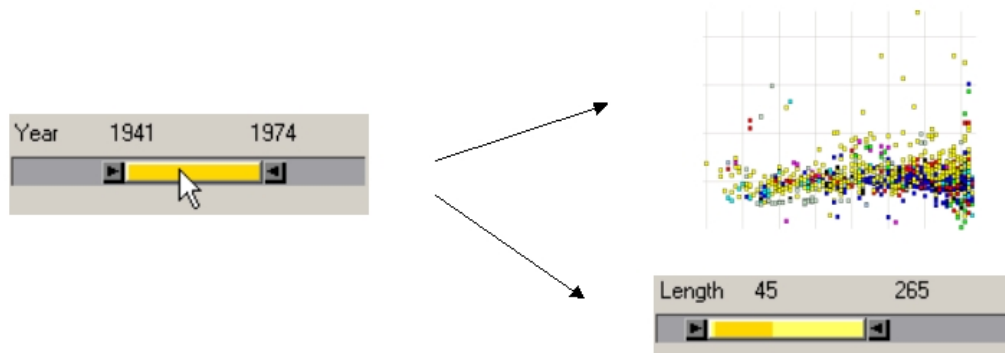


Abbildung 11 - Reglerbewegungen verursachen Ausgabe und Auswahleinschränkung

Eine wirklich enge Kopplung zwischen Ein- und Ausgab erfahren Dynamic Queries Systeme, wenn an einem Steuerelement eine Auswahl getroffen wird, eine entsprechende Ausgabe erfolgt und zudem ein anderes Steuerelement auf den Auswahlbereich beschränkt wird, der aufgrund der aktuellen Einstellung des ersten Steuerelements sinnvoll ist. In Abbildung 11 geschieht dies an zweiseitigen Schieberegler, mit denen der Anwender keinen einzelnen Wert sondern eine Spanne von Werten auswählen kann. Dazu kann er den Regler verschieben und die Länge des Reglers anpassen, indem er die Pfeile an den Rändern in die eine oder andere Richtung bewegt.

Wie in der Abbildung angedeutet, bewirkt das Verschieben des Reglers für das Jahr, der hier eine Spanne von 1941 bis 1974 abdeckt, das Verschieben der Ausgabe auf die Daten, die dieser zeitlichen Spanne entsprechen. Gleichzeitig wird der Schieberegler für die Länge angepasst, der anfangs eine beliebige Auswahl zugelassen hat, aber durch die Auswahl der Jahresspanne nur noch Längen von 45 bis 265 enthält, da der im Zeitraum von 1941 bis 1974 verfügbare Datenbestand nur Elemente enthält, die in diesem Längenbereich liegen.

Erfolgen diese Regleranpassungen auch für alle anderen Steuerelemente, dann kann der Anwender nur noch eine sinnvolle Auswahl treffen, für die Daten vorhanden sind. Entsprechend erfolgt nur dann eine leere Ausgabe, wenn keine Daten beispielsweise für den Randbereich der Steuerelemente gefunden werden aber nicht durch eine unmögliche Kombination der Regler. Zudem wird mit solchen wechselseitigen Einschränkungen der Eingabelemente eine Reduktion der je nach Anwendungsbereich sehr großen Auswahlmengen erreicht. Dadurch wird die Oberfläche der Software übersichtlicher und der Anwender kann schneller zu seinen Suchzielen gelangen.

Zum schnelleren Erzielen von Suchanfragen trägt auch das Verwenden von standardmäßigen Einstellungen beim Programmstart und die Möglichkeit, aus einer Menge von sinnvollen und interessanten Datenausschnitten zu wählen, bei. Beim Programmstart ist es beispielsweise sinnvoll, den gesamten Datenbestand darzustellen, damit der Anwender eine Übersicht vorfindet. Voraussetzung ist jedoch, dass die gewählte Darstellungsform dies aufgrund der hohen Datenmengen und beschränkten Ausgabefläche zulässt. Ebenfalls sind Presets sinnvoll, die eine Reihe von hervorgehobenen Ausgaben enthalten. Umgesetzt wird das durch unterschiedliche Konstellationen von Einstellungen der Steuerelemente, die in Presets gespeichert werden. Wählt der Anwender ein solches Preset, erhält er die entsprechenden Daten ausgegeben und die Steuerelemente bewegen sich in die der Ausgabe entsprechenden Einstellungen. Damit hat er je nach Anwendungsbereich bereits sein Suchziel erreicht. Ist dem nicht so, kann er auch von den Preseteinstellungen ausgehend durch eigene Veränderungen der Regler – durch eigenes Browsen – seine Suchanfragen durchführen. Sowohl bei der Verwendung von Starteinstellungen als auch bei Presets wird der Benutzer sofort mit Erfolgserlebnissen versorgt. Er führt nur wenige Aktionen zur Auswahl des Presets durch und erhält anschließend die entsprechende

Ausgabe. Dadurch wird die Einarbeitungszeit in Softwaresysteme erheblich reduziert und der Anwender wird durch schneller gefundene Suchergebnisse motiviert.

In den vorangegangenen Absätzen wurden Auswirkungen von Benutzeraktionen auf Eingabeelemente betrachtet. Die Zeichenfläche oder das Starfield Display waren nur starre Ausgabebereiche, die keine Interaktivität zugelassen haben. Ähnlich wie bei einem Webbrowser, bei denen der Anwender auf Hyperlinks klicken kann und Aktionen auslöst – in der Regel das Einblenden einer neuen Seite – ist es auch bei Dynamic Queries denkbar, die Zeichenfläche zu aktivieren und Interaktion zuzulassen.

Beim FilmFinder wird diese Idee angewandt, indem jeder der auf dem Starfield Display ausgegebenen und einen Film repräsentierenden Punkte anwählbar ist. Wählt der Benutzer einen dieser Punkte, erhält er in einem aufklappenden Fenster Detailinformationen zum jeweiligen Film (Details-on-demand). Damit wird eine ebenfalls enge Kopplung zwischen Eingabe und Ausgabe gemäß Tight Coupling realisiert, hier nur in anderer Richtung von Ausgabe zu Eingabe.

5 Schlusswort

Dynamic Queries sind ein faszinierendes Verfahren, um Licht in staubige Datenbestände zu bringen. Besonders, wenn in einer Anwendungsumgebung bislang nur traditionelle Abfragemethoden zur Verfügung standen, sind Visuelle Abfragen eine deutliche Bereicherung, um schnelles Auffinden von Daten zu realisieren. Bei aller positiven Betrachtung dieser neuen Ideen für Oberflächengestaltung von Suchsystemen bestehen auch einige Problemfelder. So sind Dynamic Queries für bestimmte Anwendungsbereiche gut geeignet, in manchen Bereichen aber nur bedingt einsetzbar. Beispielsweise sind Textretrievalsysteme nur in gewissen Grenzen mit Dynamic Queries zu realisieren. Man könnte zum Beispiel ein Highlighting der gefundenen Worte im Text vornehmen. Für Problemfälle mit Ausgaben in Diagrammform eignen sich visuelle Abfragen aber deutlich besser. Ein allgemeingültiges Verfahren zum Suchen von Daten sind Dynamic Queries daher nicht.

Die Anforderungen an Computersysteme sind andere als bei traditionellen Datenbanksystemen. Es wird mehr Wert auf eine möglichst schnelle Veränderung der Datenrückgabe gelegt als auf die komplette Ausgabe von Daten. Die verwendeten Algorithmen müssen daher auf schnelle Anfrage und kleine Schrittweiten optimiert werden. Je nach Anwendungsfall sind auch angepasste Datenstrukturen nötig.

Anders als bei traditionellen Datenbankabfragen wo Standards und Programmierbibliotheken (wie ODBC, JDBC, SQL, RDMBS, OODBMS) für die Entwicklung von Softwaresystemen vorliegen, sind Dynamic Queries auf den Ideen weniger Personen und Systeme aufbauend. Es fehlen daher Entwicklungswerkzeuge für die Produktion von Software mit Dynamic Queries. Ähnlich wie bei Data Mining Anwendungen fehlt Standardsoftware und jeder Anwendungsbereich erfordert neue und erhebliche Kreativität für eine angemessene Umsetzung. Falls eine wachsende Verbreitung von Dynamic Queries Systemen einsetzen sollte, wird es auch hier bequeme Methoden, Verfahren und Werkzeuge geben.

Allerdings bestehen auch fundamentale Probleme von Dynamic Queries. So ist immer ein höherer Hardwarebedarf als bei konventionellen Datenbanksystemen nötig, da nicht nur die Daten ausgegeben sondern auch eine Visualisierung erfolgen muss. Hier wird sicherlich die stetige Verbesserung von Hardware und Algorithmen Lösungen anbieten. Für die Beschränkung auf visuelle Ausgabeformen bedarf es jedoch anderer Denkansätze, da beispielsweise sehbehinderte Personen vom Visuellen Abfragen vorerst ausgeschlossen sind. Hierfür sind die aktuellen Verfahren also nicht zu gebrauchen, traditionelle Darstellungsformen können leichter für behinderte Menschen modifiziert werden.

Grundsätzlich besteht auch die Gefahr, dass Sachverhalte aufgrund ihrer visuellen Darstellung zu oberflächlich behandelt werden, da Bilder unter Umständen weniger intensiv wahrgenommen werden als andere Darstellungsformen von Problemen wie Text, Zahlen oder Formeln. Die Benutzung von Dynamic Queries darf dementsprechend nicht dazu führen, dass wichtige Datenbestände nur noch flüchtig durchflogen werden. Stattdessen muss auch mit diesem neuen System eine Interpretation erfolgen.

Dennoch bieten Dynamic Queries die geschilderten klaren Vorteile, machen einfach Spaß und dürften wohl immer häufiger in den verschiedensten Anwendungsbereichen implementiert werden.

6 Anhang

6.1 Quellenverzeichnis

(1) **“Readings In Information Visualization Using Visions To Think”**

Stuart K. Card, Jock D. Mackinlay, Ben Shneiderman, Verlag: Morgan Kaufmann 1999, ISBN 1-55860-533-9

Das Buch, aus dem die Handouts für die folgenden Artikel entnommen wurden.

(2) **“Dynamic Queries For Visual Information Seeking”**

Shneiderman, 1994 (Handout Seite 236-243 und IEEE Software 11, 6 (1994), 70-77.)

In seinem Artikel beschreibt Shneiderman die Idee des visuellen Suchens anhand unterschiedlicher Beispiele. Er vergleicht dabei insbesondere konventionelle, textbasierte Datenbankabfragen mit den visuellen Dynamic Queries und stellt Vor- und Nachteile heraus.

Umfragen mit Testanwendern geben eine Idee von der Leistungsfähigkeit visueller Abfragetechniken. Zudem nimmt Shneiderman einen Ausblick auf optimierungswürdige Aspekte vor.

(3) **“Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays”**

Ahlberg, Shneiderman, 1994 (Handout Seite 244-250)

In dieser Abhandlung wird auf grundlegende Konzepte der visuellen Informationssuche eingegangen und diese anhand des Projekts FilmFinder beschrieben, mit dem visuelles Suchen in einer Filmdatenbank ermöglicht wird.

Es werden Dynamic Queries als Abfragen von Datenbeständen durch visuelle Interaktion eingeführt, genauso wie das Abbilden von Daten auf zweidimensionale Ebenen mit Starfield Displays. Außerdem wird das Tight Coupling, die Rückkopplung bereits angezeigter Daten auf die Eingabefelder, beschrieben.

(4) **„Data Visualization Sliders“**

Stephen G. Eick, AT&T Bell Laboratories, eick@research.att.com (Handout Seite 251-252)

Dynamic Queries sind schnelle Änderungen von Informationsvisualisierungen, die der Anwender per Interaktion mit Steuerelementen in Echtzeit kontrolliert. Traditionelle Steuerelemente von Benutzeroberflächen wie Buttons, Scrollbars, Texteingabefeldern oder Listen können erweitert werden, um schnellere und intuitivere Benutzerführungen zu erlangen. Der hier vorgestellte Data Visualization Slider erweitert herkömmliche Scrollbars um präzisere und umfangreichere Selektionen vornehmen zu können.

(5) **„Enhanced Dynamic Queries via Movable Filters“**

Ken Fishkin, Maureen C. Stone, Xerox PARC, {fishkin, stone}@parx.xerox.com (Handout Seite 253-259)

Die Autoren stellen ein intuitiv zu nutzendes Prinzip vor, wie man die Verknüpfungsmöglichkeiten einer booleschen Abfragesprache mit grafischen Mitteln realisiert. Ihre Methode wird Movable Filter oder Magic Lens genannt, da ein oder mehrere, mit individuellen Abfrageeinschränkungen versehene Filter über

Informationskarten gelegt und die dargestellten Informationseinheiten je nach Wertigkeit Ihrer Attribute dargestellt werden.

(6) Software „Spotfire Pro“

Spotfire Pro 3 Demo for Windows, © 1996-98 Spotfire AB, Version 3.3.4 (siehe <http://www.spotfire.com>)

Die Software der gleichnamigen Firma Spotfire, die von Christopher Ahlberg gegründet wurde, stellt Visualisierungsvorlagen für verschiedene Anwendungsbereiche aus Wirtschaft, Industrie und Geografie zur Verfügung. Auch der vormals auf Unix-Plattformen entwickelte FilmFinder ist hier als Windows-Implementierung enthalten.

(7) “Dynamic queries, starfield displays, and the path to Spotfire”

Ben Shneiderman, February 4, 1999 (siehe <http://www.cs.umd.edu/hcil/spotfire/>)

Shneiderman fasst seine Bemühungen, Entwicklungen und Erfolge bei der Suche nach bedienungsfreundlichen Programmoberflächen zusammen. Es wird ein Überblick über die seit Anfang der neunziger Jahre bis 1999 erzielten Fortschritte gegeben und einiges an Personeninformationen veröffentlicht, was ein interessantes Bild von der relativ kleinen Gemeinde von Wissenschaftlern und Studenten bietet, die sich mit dem Thema Dynamic Queries beschäftigen.

6.2 Index

A

Abfrage 3, 4, 5, 6, 7, 8, 9, 10, 14, 15
 Algorithmen..... 9, 10
 Alphaslides..... 6, 8, 9
 Anfrage 3, 4, 5, 6, 7
 Anwender..... 3, 4, 5, 6, 7, 9, 11
 Anwendung..... 5, 6, 7, 11

B

Benutzer..... 5, 10, 11, 14
 Benutzerinterface 3, 5, 7
 Browsen..... 5, 7, 9, 14, 15
 Button 5, 6, 9

D

Datenstruktur 9, 10
 Details-on-demand..... 6, 8, 11, 14
 Dynamic Queries 1, 3, 4, 5, 6, 7, 8, 9, 11, 13, 14,
 15, 16, 17

E

Echtzeit 10

F

FilmFinder 6, 7, 8, 9

I

Interaktion..... 3, 10, 11, 13, 17

K

Koordinatensystem 3, 6, 7, 9, 11, 13

M

Mapping..... 11, 12, 13

O

Oder-Verknüpfung..... 8

P

Periodensystem 12
 Preset..... 14, 16
 Psychologischer Moment..... 9, 10
 Punktgraph 7, 11, 13

R

Reaktionszeit..... 6, 9, 10
 Reversibilität..... 15

S

Softwaredesign..... 14, 15
 Spontane Reaktion 9
 SQL 4, 6, 15
 Starfield Displays..... 3, 7, 8, 9, 11, 13, 14
 Steuerelement... 5, 6, 7, 8, 9, 10, 11, 14, 15, 16,
 19

T

Textverarbeitung 14, 15
 Tight Coupling 3, 5, 9, 14, 15

U

Und-Verknüpfung 8
 unit task..... 9

V

Videos-on-demand 7
 Visualisierung 3, 5, 6, 7, 11, 13
 visuell..... 4, 5, 6, 7, 9, 10, 11

X

XQL 4, 6