

Dynamische Visualisierung von Geschäftsdaten im Business Intelligence-Prozess mit Hilfe von Rich Internet Applications

Diplomarbeit

zur Erlangung des Grades eines Diplom-Informatikers
im Studiengang Computervisualistik

vorgelegt von
Benjamin Schmidt

Erstgutachter: Prof. Dr.-Ing. Stefan Müller
(Institut für Computervisualistik, AG Computergraphik)
Zweitgutachter: Dr. Jörg Reinhardt
(Business Manager, Altran CIS GmbH)

Koblenz, im Februar 2008

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ja Nein

Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.

.....
(Ort, Datum)

.....
(Unterschrift)

Danksagung

Der Weg bis zum Abschluss meines Studiums war spannend und interessant, aber auch manchmal steinig und beschwerlich - nicht nur für mich, sondern auch für die Menschen, die mich begleitet haben. Diesen Personen gebührt mein Dank.

Zuerst einmal gehören hierzu sicherlich die Betreuer meiner Diplomarbeit. Seitens Altran CIS sind dies Dr. Jörg Reinnarth und Magdalena Biadala-Sahingöz, die mir beide stets hilfsbereit und unbürokratisch den benötigten Rückhalt gegeben haben, sowie Markus Ehl, mit dem ich die Kooperation beginnen durfte.

Als ich mit dem bereits fertigen Konzept meiner Diplomarbeit zu Prof. Dr. Stefan Müller von der Universität Koblenz- Landau ging, war er sofort entschlossen mir die Möglichkeit zu geben, meine Arbeit bei ihm zu schreiben. Ich kann ihm dafür nicht genug danken.

Besonders am Herzen liegt mir natürlich, dass meine Familie versteht, wie sehr ich ihre Unterstützung während all der Jahre zu schätzen weiß. Meine Eltern boten mir fortwährend eine liebevolle Umgebung, und nutzten alle Gelegenheiten mich zu fordern und zu fördern. Mit ihrer Hilfe konnte ich mich laufend weiterentwickeln – in dem Wissen, dass sie für mich da sein werden, wenn ich sie brauche.

Ansporn dazu, ständig weiter zu machen, gab mein Leben lang auch besonders meine Großmutter, und sei es, weil ich immer sicher war, dass sie auf jeden meiner Erfolge unglaublich stolz ist.

Außer meinen Verwandten und meinen Betreuern gibt es noch einen Menschen, dem ich ganz besonderen Dank schulde. Meine Freundin Johanna Krichling hat mir in den letzten viereinhalb Jahren die nötigen Anstöße gegeben, wenn ich mich in einer Sackgasse sah, mich inspiriert und mich mit ihrer Liebe, ihrem Wissen und Können so weit gebracht, dass ich mich nun hoffentlich bald Diplom- Informatiker (CV) nennen darf.

Institut für Computervisualistik
AG Computergraphik
Prof. Dr. Stefan Müller
Postfach 20 16 02
56 016 Koblenz
Tel.: 0261-287-2727
Fax: 0261-287-2735
E-Mail: stefanm@uni-koblenz.de



UNIVERSITÄT
KOBLENZ · LANDAU

Fachbereich 4: Informatik

Aufgabenstellung für die Diplomarbeit
Benjamin Schmidt
(Matr.-Nr. 201 110 095)

Thema: Dynamische Visualisierung von Geschäftsdaten im Business Intelligence- Prozess mit Hilfe von Rich Internet Applications

Im Laufe der Zeit fallen in einem Unternehmen große Mengen von Daten und Informationen an. Die Daten stehen im Zusammenhang mit unternehmensinternen Vorgängen, mit dem Marktumfeld, in dem das Unternehmen positioniert ist, und auch mit den Wettbewerbern. Sie sind vielfältiger Art, normalerweise inhomogen und aus verteilten Datenquellen zu beziehen.

Um in dieser Flut von Daten die Übersicht zu behalten, die Menge an Informationen effektiv für das Unternehmen nutzbar zu machen, vor allem auch nachhaltig für kommende Entscheidungsfindungen, müssen die Daten analysiert und integriert werden. Diese Optimierung der Entscheidungsfindung durch Zugang zu Informationen, deren Analyse und Auswertung wird häufig unter dem Begriff "Business Intelligence" zusammengefasst.

Der Wert der vorhandenen Informationen hängt stark von dem erwähnten Zugang und einer ausdrucksstarken Repräsentation ab. RIA-Technologien ermöglichen eine einfache Verfügbarkeit der verarbeiteten Geschäftsdaten über Inter- und Intranet ohne große Anforderungen an ihre Nutzbarkeit zu stellen. Sie bieten zudem spezialisierte leistungsfähige und in großem Maße programmierbare Visualisierungstechniken.

Die Diplomarbeit soll zwei Schwerpunkte haben. Zum Einen wird sie sich mit Arten der Informationsvisualisierung im Allgemeinen und deren Eignung für Geschäftsdaten beschäftigen. Der Fokus liegt hierbei auf Daten und Informationen in Management-Informationsberichten. Ziel ist eine Visualisierungsform, die es dem Nutzer ermöglicht, die zu kommunizierenden Informationen effizient auszuwerten. Zum Anderen untersucht die Diplomarbeit die Vor- und Nachteile des Einsatzes von RIAs. Der Implementierungsteil umfasst eine RIA als "Proof of Concept", deren Hauptaugenmerk auf eine dynamische Interaktion und optimierte Informationsvisualisierung gerichtet sein soll.

Die Diplomarbeit wird bei der Altran CIS in Koblenz durchgeführt.

Betreuer: Markus Ehl (De Simone & Osswald AG)

Koblenz, den 27. Juni 2007

- Prof. Dr. Stefan Müller-

Inhaltsverzeichnis

Danksagung	i
Abkürzungsverzeichnis	ix
1 Einleitung	1
1.1 Motivation	1
1.2 Firmenprofil	1
1.3 Aufgabenstellung	2
1.4 Aufbau der Arbeit	2
2 Grundlagen	5
2.1 Business Intelligence	5
2.1.1 Definition	5
2.1.2 Einsatzgebiete	6
2.2 Rich Internet Applications	6
2.2.1 Definition	6
2.2.2 Geschichte	7
2.2.3 Arten	8
2.3 Informationsvisualisierung	9
2.3.1 Definition	10
2.3.2 Geschichte	11
2.3.3 Konzepte	12
2.3.4 Formen	32
3 Realisierung	39
3.1 Anforderungen	39
3.2 Grafische Benutzungsoberfläche	40
3.3 Adobe Flash / Flex	43
3.3.1 MXML	45
3.3.2 ActionScript 3	45
3.3.3 Flex- Schnittstellen	47
3.4 Architektur	48
3.5 Interner Programmablauf	49
3.6 Bedienung	54
4 Bewertung der Anwendung	60

5 Zusammenfassung und Ausblick	65
5.1 Zusammenfassung	65
5.2 Ausblick	67
A Klassendiagramme	71
B Fragebogen	73
Seite 1	73
Seite 2	74
Seite 3	75

Abbildungsverzeichnis

1	Schriftliche Division	10
2	da Vinci: Schädelstudie	11
3	Visualisierungsprozess, schematisch	13
4	Kodierungen, nach Datentypen geordnet	15
5	Ungültige Kodierung (Ausdrückbarkeitskriterien)	16
6	Datentabelle mit Metadaten	17
7	Layering und Separation	18
8	Maximierung des Datentinte-Anteils (Boxplot)	19
9	Maximierung des Datentinte-Anteils (Scatterplot)	20
10	Alignment, Folding, Recursion, Overloading	21
11	Treemap	22
12	Simultankontrast	22
13	Hermannsches Gitter, Szintillationsgitter	23
14	Machsches Band	24
15	Farbfehlsichtigkeit und Sättigung	25
16	Farbfehlsichtigkeit und Form	26
17	Horizontlinie in Grafiken	26
18	Farbfehlsichtigkeit und Form	28
19	Farbfehlsichtigkeit und Form	29
20	Gestaltgesetze: Ähnlichkeit, Gemeinsame Region	30
21	Gestaltgesetze: Geschlossenheit, Gute Fortsetzung	30
22	Gestaltgesetze: Gute Gestalt, Nähe	31
23	Gestaltgesetze: Verbundenheit	31
24	Box- and- Whiskers Diagramm	33
25	Bulletgraph, Elemente	34
26	Bulletgraph, Varianten	35
27	Nomogramm	36
28	Logarithmentafel	37
29	Kreisdiagramm	38
30	Liniendiagramm und Cycle- Chart	38
31	Navigation in der Anwendung	40
32	Haupt- Sicht der Anwendung	41
33	Detail- Sicht der Anwendung	42
34	Analysesicht der Anwendung	43
35	Schema der Flex Runtime Architektur	44
36	Beispiel: MXML nach SWF- Übersetzung	47
37	DataProvider.xml	50
38	deepCopy- Funktion	51
39	setStartAndEndTime- Funktion	52
40	Popup- Fenster der Analysesicht	53
41	Analysesicht der Anwendung	54
42	convertArrayCollectionToSet- Funktion	55

43	testCorrectnessOfDataInfosXML- Funktion	57
44	calcLinRegression- Funktion	58
45	Trenderkennung durch logarithmische Achse	59
46	Anpassung des angezeigten Bereichs	59
47	Klassen der Sparkline- Komponente	71
48	Auszug des Klassendiagramms	72

Abkürzungsverzeichnis

AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
AS3	ActionScript 3
BI	Business Intelligence
CRM	Customer Relationship Management
CSS	Cascading Style Sheet
CSV	Comma Separated Values
GOMS	Goals, Operators, Methods, Selection Rules
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IDE	Integrated Development/Design Environment
RIA	Rich Internet Application
SDK	Software Development Kit
SWF	ShockWave Flash / Small Web Format

1 Einleitung

Im Folgenden soll ein Überblick über die Ziele gegeben werden, die diese Arbeit verfolgt. Es wird geklärt, welche Notwendigkeiten hierzu geführt haben, und wie die Problemstellungen angegangen werden.

1.1 Motivation

Die Optimierung der Entscheidungsfindung im Geschäftsumfeld durch Zugang zu Informationen, deren Analyse und Auswertung wird häufig unter dem Begriff Business Intelligence (Kap. 2.1) zusammengefasst. Der Wert der vorhandenen Informationen hängt dabei stark von dem erwähnten Zugang und einer ausdrucksstarken Repräsentation ab.

Um eine einfache Verfügbarkeit der verarbeiteten Geschäftsdaten über Inter- und Intranet zu ermöglichen, bieten sich Rich Internet Applications mit spezialisierten, leistungsfähigen und in hohem Maße programmierbaren Visualisierungstechniken an, ohne große Anforderungen an ihre Nutzbarkeit zu stellen.

Diese Diplomarbeit wird in Zusammenarbeit mit dem Business- und IT-Beratungsunternehmen Altran CIS GmbH (im Folgenden Altran CIS genannt) geschrieben. Die Anforderungen an die zu erstellende Anwendung wurden im Vorfeld abgestimmt. Weiterhin stellte Altran CIS generierte Testdaten zur Verfügung.

1.2 Firmenprofil

Als Mitglied der französischstämmigen Altran Group besteht Altran CIS international aus mehr als 5000 Mitarbeitern, davon etwa 150 deutschlandweit an sechs Standorten. Die Hauptgeschäftsfelder des Unternehmens sind Business Intelligence (sh. dazu auch Kap. 2.1) und Customer Relationship Management Lösungen zusammen mit der damit verbundenen IT-Beratung, Entwicklung von Architekturen, bis hin zur Implementierung. Im Bereich Business Intelligence bietet Altran CIS Dienstleistungen wie

Strategie- und Business- Consulting, Datawarehouses und ETL¹- Prozesse und Datamining, sowie Reporting und Dashboards.

1.3 Aufgabenstellung

Die Diplomarbeit soll zwei Schwerpunkte haben. Zum Einen wird sie sich mit Konzepten und Formen der Informationsvisualisierung beschäftigen, wobei der Fokus hierbei auf abstrakten Geschäftsdaten und ihrer Darstellung in verschiedenen Detailstufen liegt. Ziel ist eine Visualisierungsform, die es dem Nutzer ermöglicht, die zu kommunizierenden Informationen effizient auszuwerten.

Zum Anderen untersucht die Diplomarbeit die Vor- und Nachteile des Einsatzes von Rich Internet Applications (RIAs, Kap. 2.2). Als „Proof of Concept“ wird eine RIA auf Grundlage von Adobe Flex 2 entwickelt, die die recherchierten Konzepte und Techniken umsetzt.

1.4 Aufbau der Arbeit

In Kapitel 2 wird diese Arbeit die Grundlagen der relevanten Themen behandeln. Hierzu gehören die Bereiche Informationsvisualisierung, Rich Internet Applications und Business Intelligence. Die Arbeit wird im Kapitel 2.1 erklären, was allgemein unter dem Begriff „Business Intelligence“ (BI) verstanden wird, und welche Einsatzgebiete es generell für BI- Techniken gibt.

Die rasante Entwicklung der letzten 10 Jahre im Bereich der IT- Technologie, speziell was das Internet und die Kapazitäten der Personal Computer angeht, führte bei netzwerkbasierenden Anwendungen zu einer weiteren Verbreitung der Rich Client Technologie. In Kapitel 2.2 wird diese Tendenz genauer vorgestellt. Die Möglichkeiten von Rich Internet Applications werden untersucht, und es wird eine Unterscheidung in zwei Hauptarten von RIAs auf Basis ihrer sehr speziellen Eigenschaften vorgeschlagen.

Besonderer Wert wird im Grundlagen- Kapitel auf das Thema Informationsvisualisierung gelegt. Es wird erläutert, wieso eine wohlüberlegte

¹*Extract, Transform, Load*: Vorgehensmodell um heterogene Daten aus mehreren Quelldatenbanken in einer Zieldatenbank zu integrieren

Visualisierung aufgrund der heutigen Mengen an Informationen, mit denen wir konfrontiert sind, unumgänglich ist und was beachtet werden muss um gute Ergebnisse zu erreichen. Schließlich werden die wichtigsten Formen aufgezeigt, die das Fundament der Informationsvisualisierung auch im Bereich der geschäftlichen Daten bilden. Es wird jeweils versucht zu bestimmen, für welche Zwecke sich die einzelnen Arten besonders eignen und welche Probleme sich stellen.

Die Ergebnisse des Grundlagenkapitels fließen in die entwickelte Anwendung ein. Daher wird in Kapitel 3 dokumentiert, wie die Rich Internet Application, die im Verlauf dieser Arbeit zu erstellen war, realisiert wurde. In Zusammenarbeit mit Altran CIS wurden grundlegende Anforderungen erhoben. Diese führten zu einem Entwurf der grafischen Benutzungsoberfläche und einer Entscheidung über die zu verwendende Architektur. Die Werkzeuge Adobe Flash und Flex werden vorgestellt, und es wird erklärt, weshalb sich diese Technologien besonders gut eignen. Anhand eines möglichen Nutzungsdurchlaufes werden die Interna der Anwendung und ihr Zusammenspiel genauer beschrieben.

Zur Überprüfung in welchem Umfang die Prinzipien einer guten Informationsvisualisierung in der entwickelten RIA umgesetzt wurden, diente ein auf entsprechenden Hypothesen basierender Fragebogen, der von Altran CIS-Mitarbeitern ausgefüllt wurde, und dessen Ergebnisse in Kapitel 4 ausgewertet werden.

Schließlich werden die im Verlauf dieser Diplomarbeit gewonnenen Erkenntnisse zusammengefasst und zusammen mit einem Ausblick auf mögliche Änderungen und Erweiterungen der Rich Internet Application in Kapitel 5 präsentiert. Viele Ideen und Vorschläge konnten aufgrund des zeitlichen Rahmens der Arbeit und einiger Einschränkungen auf technischer Seite nicht umgesetzt werden. Ein Überblick soll hier gegeben werden.

2 Grundlagen

Dieses Kapitel wird die grundlegenden Begriffe klären, die zum Verständnis der Arbeit notwendig sind. Die drei Gebiete von hauptsächlichem Interesse, Business Intelligence, Rich Internet Applications und Informationsvisualisierung, werden mit ihrer Geschichte, ihren Eigenschaften und Einsatzmöglichkeiten genauer untersucht.

2.1 Business Intelligence

Im Laufe der Zeit fallen in einem Unternehmen große Mengen von Daten und Informationen an. Die Daten stehen im Zusammenhang mit unternehmensinternen Vorgängen, mit dem Marktumfeld, in dem das Unternehmen positioniert ist, und auch mit den Wettbewerbern. Sie sind vielfältiger Art, üblicherweise inhomogen und aus verteilten Datenquellen zu beziehen.

Der Zuwachs an Datenmengen führt ohne eine strategische Datenhaltung und Informationserschließung nicht wie erwartet zu einem Mehr an Information sondern zu einer Mangelsituation. Um in der Menge von Daten die Übersicht zu behalten und die Informationen effektiv für das Unternehmen nutzbar zu machen – vor allem auch nachhaltig für kommende Entscheidungsfindungen – müssen die Daten analysiert und integriert werden.

2.1.1 Definition

Den Begriff der Business Intelligence (BI) prägte das Beratungsunternehmen Gartner² im Hinblick auf die Fähigkeit zur Optimierung der Entscheidungsfindung durch Zugang zu Informationen, deren Analyse und Auswertung (Ramsdale, 2001, S. 5).

Die Prozesse der Business Intelligence bestehen nach Grothe (2000) im Wesentlichen aus der Bereitstellung der Datengrundlage (z.B. Data Warehouses), der Erschließung von Zusammenhängen (z.B. Data Mining) und schließlich der Kommunikation der Ergebnisse.

²Damals „Gartner Group“: <http://www.gartner.com/>

Um einen vereinfachten und leicht verständlichen Definitionsvorschlag einzuführen, soll hier die Studie (Hillringhaus und Kedzierski, 2004) der Fachhochschule Köln herangezogen werden, in der nach dem Verständnis der Unternehmen von Business Intelligence gefragt wurde. Dort ergab sich die überwiegende Meinung, dass BI ein Prozess sei, „durch den Geschäftsdaten in Informationen zur Unternehmenssteuerung umgewandelt werden“.

2.1.2 Einsatzgebiete

In einem von Gartner veröffentlichten Bericht wird die folgende Rangfolge der Einsatzzwecke von BI aufgestellt (Willen, 2002).

1. Corporate/Business Performance Management
2. Optimierung der Kundenbeziehungen (CRM³), Überwachung der Geschäftsaktivitäten, Unterstützung herkömmlicher Entscheidungsfindung
3. Standalone Anwendungen für spezifische Arbeitsabläufe oder strategische Entscheidungen
4. Management- Berichte

2.2 Rich Internet Applications

Mit der Verbreitung von Breitband- Internet- Anschlüssen und leistungsfähigen Heim- Computern entwickelte sich in den letzten Jahren die Möglichkeit netzwerkfähige Anwendungen statt in Thin Client- Technologie als Rich Clients auf Servern bereitzustellen. Im Folgenden werden verschiedene Varianten dieser Art von Programmen vorgestellt.

2.2.1 Definition

Rich Internet Applications sind Web- Anwendungen mit grafischer Benutzungsoberfläche, die auf einem Server vorgehalten werden, jedoch als *Thick Clients* lokal im Webbrowser ausgeführt werden, Daten zwischen Client

³*Customer Relationship Management*

und Server kommunizieren und dabei im Gegensatz zu traditionellen Webseiten weitgehende Eigenschaften von Desktop- Anwendungen aufweisen (Fraunhofer Institut für Angewandte Informationstechnik, 2006).

2.2.2 Geschichte

Die erste Erwähnung des Begriffs „Rich Internet Application“ (RIA) findet sich in einem Whitepaper des Unternehmens Macromedia aus dem Jahre 2002 (Allaire, 2002). Die Wurzeln der asynchronen Kommunikation der RIA- Technologie AJAX liegen jedoch noch ein halbes Jahrzehnt davor bei dem IFRAME- (Internet Explorer 3, 1996) und dem Layer- Syntaxelement (Netscape 4, 1997). Mit Microsoft Remote Scripting (1998) gab es zudem bereits die Möglichkeit Skripte lokal im Internetbrowser auszuführen.

In den Anfängen des Internets lag die Rechenleistung vor allem bei den Servern, so dass die Client- Rechner möglichst wenig belastet werden sollten. Die klassische HTML- Seite spiegelt diese Umstände wider. Leistungsstarke Computer wurden immer erschwinglicher, und die Verbreitung des Internets auch im privaten Nutzungsbereich nahm stark zu. Dies führte wiederum zu einer großen Belastung der Server in der Kommunikation, die vermehrt Anfragen verarbeiten und beantworten mussten.

Heute entlasten RIAs durch Ausnutzung der lokalen Rechenleistung die Server und nutzen zugleich die Möglichkeiten moderner Heimcomputer. Mit der Verbreitung schneller Internetanbindungen und dem vermehrten Bedürfnis an Flexibilität und an steter Verfügbarkeit können RIAs auch als Lösung für Software-on-Demand gesehen werden.

Das heisst zum Beispiel im Unternehmensumfeld, dass ein Software-Produkt nicht einzeln auf dem Computer jedes Mitarbeiters installiert und eingerichtet werden muss. Die Anwendung wird auf einem Server vorgehalten und bei Bedarf über das Netzwerk geladen und gestartet. Durch Laden nur der gerade benötigten Komponenten des Programms, späteres Streaming im Hintergrund und Kompression kann hierbei schnell mit der Nutzung begonnen werden. Gibt es ein Software- Update oder eine Erweiterung, muss diese nur ein Mal auf dem Server angewendet werden und kommt mit sofortiger Wirkung allen Nutzern zugute.

2.2.3 Arten

Technologisch werden RIAs in skript- basierte und plugin- basierte Ansätze unterschieden. Skript- basierte Anwendungen arbeiten mit der üblichen HTML- Sicht, erweitert um skript- gesteuerte Aktionen, Interaktionen und Darstellungen. Als bekannteste skript-basierte Lösung ermöglicht es AJAX, mittels JavaScript asynchrone HTTP- Anfragen an den Server zu stellen und nur die benötigten Teile einer HTML- Seite neu zu laden. Ein wichtiger Vorreiter bei AJAX- Anwendungen ist die Google Inc. mit Produkten wie Groups⁴ oder Maps⁵. Die Anpassung an die verschiedenen Browser stellt eine große Herausforderung bei AJAX- Anwendungen dar, da eine generelle Browserkompatibilität nicht gegeben ist. Zudem hegen viele Nutzer aus Sicherheitsgründen Vorbehalte gegenüber JavaScript und deaktivieren diese Funktionalität. Dies macht jedoch eine Nutzung von AJAX- Anwendungen unmöglich.

Das Ausgabeformat der am meisten verbreiteten plugin- basierten Lösungen ist SWF⁶ von Adobe. Mit Hilfe des Flash Player- Zusatzes, der für alle wichtigen Browser kostenlos zum Download angeboten wird, können die vorkompilierten Anwendungen genutzt werden. Flash Player 9 unterstützt mit seiner von Grund auf neu entwickelten virtuellen Maschine nun auch Just-in-Time Kompilierung, was zu einem starken Leistungszuwachs gegenüber seinem Vorgänger führt. Laut Adobe lag die weltweite Verbreitung des Plugins unter Internet- Nutzern im Dezember 2007 bei etwa 99% (Adobe Systems, Inc., b), und selbst die jüngste Version 9 war bei mehr als 93% der Internet-Nutzer installiert (Adobe Systems, Inc., a).

Da der SWF-Standard von Adobe offengelegt wurde, sind Entwickler nicht zwangsläufig auf die kostenpflichtige Entwicklungsumgebungen Adobe Flash oder Adobe Flex Builder angewiesen. Für die Erstellung von RIAs sind die spezialisierten und kostenfrei nutzbaren SDKs⁷ von Adobe und OpenLaszlo interessant geworden. Sie vereinfachen mit angepassten XML-Dialekten die Programmierung von RIAs erheblich.

⁴<http://groups.google.de/>

⁵<http://maps.google.de/>

⁶*ShockWave Flash* oder auch *Small Web Format*

⁷*Software Development Kit*

Als Konkurrenzlösung zu Adobe Flash wurde im April 2007 von Microsoft die Browsererweiterung Silverlight vorgestellt, die .NET-Entwickler ansprechen soll und ein weiteres proprietäres Ausgabeformat nutzt. Allerdings befindet sich das Plugin noch in einem sehr frühen Stadium. Außerdem existieren momentan nur Versionen des Plugins und des APIs⁸ für die Betriebssysteme Microsoft Windows und Apple Mac OS. Eine Umsetzung für Mono zur Nutzung Silverlights zum Beispiel unter Linux ist von der Nutzergemeinschaft angestrebt. Aus diesen Gründen wird im Rahmen dieser Diplomarbeit Adobe Flex 2 eingesetzt. Details hierzu werden in Kapitel 3.3 erläutert.

Plugin- basierte Lösungen bieten dem Entwickler den großen Vorteil, dass der Inhalt, vom genutzten Browser unabhängig, immer gleich dargestellt wird. Denn, anders als bei skript- basierten RIAs, ist hier das jeweilige Plugin für das Rendern zuständig, nicht die Browser-Engine.

2.3 Informationsvisualisierung

„The power of the unaided mind is highly overrated. Without external aids, memory, thought, and reasoning are all constrained. But human intelligence is highly flexible and adaptive, superb at inventing procedures and objects that overcome its own limits. The real powers come from devising external aids that enhance cognitive abilities. How have we increased memory, thought, and reasoning? By the invention of external aids: It is things that make us smart.“

(Norman, 1993, S. 43)

Die Kapazität des menschlichen Gehirns ist stark begrenzt. Viele Informationen gleichzeitig vorzuhalten und zu bearbeiten ist nur schwer möglich. Gleichzeitig besitzt der Mensch die Fähigkeit externe Hilfsmittel zu entwickeln und sich ihrer Unterstützung zu bedienen. Hat ein Mensch z.B. solche ausgelagerten Speicher zur Verfügung, kann er die Leistung seines Gehirnes erst wirklich entfalten. Dieser Umstand führte unter Anderem zu der Erarbeitung schriftlicher Rechentechniken (Abb. 1) oder der Entstehung von Diagrammen.

⁸Application Programming Interface

$$\begin{array}{r}
 1032 / 3 = 344 \\
 - \underline{9} \\
 \quad 13 \\
 - \underline{12} \\
 \quad \quad 12 \\
 - \underline{12} \\
 \quad \quad \quad 0
 \end{array}$$

Abbildung 1: Die schriftliche Division entlastet das Gehirn des Rechnenden, da sie die Zwischenergebnisse der Aufgabe sichtbar hält und der Rechnende sich so auf die aktuellen und überschaubaren Zwischenrechnungen konzentrieren kann

Eine herausragende Gruppe externer Hilfestellungen ist grafischer Natur. Sie vermitteln entweder eine Idee („Ein Bild sagt mehr als tausend Worte“) oder – falls eine zu kommunizierende Idee noch nicht vorliegt – helfen bei ihrer Erarbeitung (Card u. a., 1999).

2.3.1 Definition

Der Begriff der Informationsvisualisierung ist keineswegs klar definiert und allgemein in einer einzigen Bedeutung anerkannt. Die Literatur bietet in erster Linie die nun folgenden Definitionsversuche.

Spence (2001) sieht Informationsvisualisierung als „Prozess um ein mentales Modell einer Datenstruktur zu formen“ oder auch als „Unterstützung der Dateneinsicht durch Mustererkennung in Displays.“. Er weist zudem darauf hin, dass bei dieser Art der Visualisierung der Fokus auf der Darstellung abstrahierter Daten liegt.

Card, Mackinlay und Shneiderman (1999) sehen „Visualisierung“ als die Nutzung computer- unterstützter, interaktiver, visueller Repräsentationen von Daten um Wissen zu verstärken. Der Zweck der Visualisierung (frei nach Richard Hamming) ist „*insight, not pictures*“.

Die Möglichkeit sich Zusammenhänge in grafischer Form darstellen zu lassen rechtfertigt also nicht allein den Prozess der Visualisierung. Der Aufwand soll nur dann durchgeführt werden, wenn dadurch eine größere Einsicht gewonnen wird.



Abbildung 2: Schädelstudie von Leonardo da Vinci, 1508

Der Schritt von der Definition der Visualisierung zur Definition der „Informationsvisualisierung“ liegt für Card in einem zusätzlichen Begriff. Er schreibt:

„[Informationsvisualisierung ist] die Nutzung computer- unterstützter, interaktiver, visueller Repräsentationen **abstrakter** Daten um Wissen zu verstärken.“

2.3.2 Geschichte

Die geschichtliche Entwicklung der Informationsvisualisierung besteht aus vielen kleinen Schritten bis in die Gegenwart. Im Folgenden sollen einige herausragende Stationen besonders beleuchtet werden.

Bereits Anfang des 16. Jahrhunderts ist es für Leonardo da Vinci selbstverständlich, Grafiken in seine Texte zu integrieren oder die von ihm entworfenen Grafiken zu annotieren, wie zum Beispiel in Abbildung 2. Hierzu verbindet er den textuell erläuterten Teil der Zeichnung durch einen Strich

mit der zugehörigen Erklärung oder schreibt diese direkt auf den entsprechenden Teil. In der Schule der Gestaltpsychologie des 20. Jahrhunderts (Kap. 2.3.3, Seite 28) wird schließlich erforscht, welche Effekte solche abstrakten Zusammengehörigkeitsanzeiger auf den Menschen haben.

Den nächsten großen Entwicklungsschritt hin zu der gegenwärtigen Ausprägung der Informationsvisualisierung macht im 18. Jahrhundert William Playfair. In seinem Werk „The Commercial and Political Atlas“⁹ (Playfair, 1786) aus dem Jahr 1786 finden sich die ersten Nutzungen von Balkendiagrammen. Playfair gilt nicht nur als Erfinder des Balkendiagramms (sh. auch Kap. 2.3.4, S. 32). Er entwarf zudem die sogenannten Kreis- oder auch Tortendiagramme (sh. auch Kap. 2.3.4, S. 36).

Bertin (1967) beschreibt als Erster vor seinem Hintergrund als bekannter Kartograf detailliert Grundelemente von Diagrammen. Ein wichtiges Thema für ihn ist die Theorie der strukturellen Objekt- Zeichen- Beziehung. Diese Theorie führt später zur Einteilung darzustellender Daten in qualitative (nominale), geordnete (ordinale) und quantitative (intervall) Daten. Aufgrund dieser Einteilung ordnet zum Beispiel Card u. a. (1999) Daten bestimmte zu nutzende Visualisierungsarten zu (sh. auch Kap. 2.3.3, S. 17).

Der wahrscheinlich einflussreichste Wissenschaftler der letzten 30 Jahre auf dem Gebiet der Informationsvisualisierung ist Edward Rolf Tufte. Eines des bekanntesten von ihm entwickelten Konzepte ist der sogenannte *Datentinte- Anteil* der Grafiken. Erläuterungen dazu sind in Kapitel 2.3.3 auf Seite 17 zu finden. Breite Anerkennung in der wissenschaftlichen Öffentlichkeit erlangte die Informationsvisualisierung schließlich durch die erste IEEE- Visualisierungskonferenz im Jahre 1990.

2.3.3 Konzepte

Visualisierung soll stets auf die effektive Kommunikation von Information abzielen. Dazu muss sowohl die Information selbst, als auch die angestrebte Nutzergruppe verstanden werden. (Shephard und Schroeder, 1995)

Der Prozess der Visualisierung (Abb. 3) nach Ware (2004) besteht im Wesentlichen aus den folgenden Schritten: Zuerst werden Daten gesammelt

⁹Vollständiger Titel: „*The Commercial and Political Atlas: Representing, by Means of Stained Copper-Plate Charts, the Progress of the Commerce, Revenues, Expenditure and Debts of England during the Whole of the Eighteenth Century*“

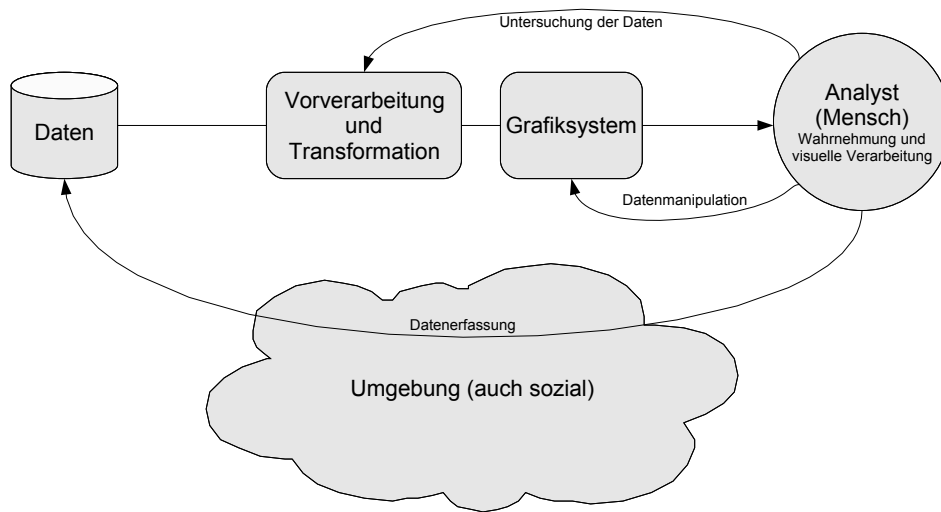


Abbildung 3: Visualisierungsprozess (schematisch), nach Ware (2004)

und gespeichert und dann in Vorverarbeitungsschritten durch Transformationen in eine verständliche Form gebracht. Eine weitere Transformation wandelt die Daten in eine an das jeweilige Ausgabemedium angepasste Form. Über das Ausgabemedium nimmt schließlich der Betrachter die Daten wahr und interpretiert sie. Dies geschieht immer auch unter sozialen und kulturellen Aspekten.

Abbildung von Daten Sollen Daten grafisch dargestellt werden, so muss ein Weg gefunden werden, den „Datenraum“ in den „Repräsentationsraum“ abzubilden (Deussen, 2002). Hierbei sind die Daten selbst zu berücksichtigen, da anhand der Art der Daten (nominal, ordinal, quantitativ) die jeweils geeigneten Visualisierungen zu wählen sind. Hinzu kommt die Überlegung, welche Visualisierungen im Zielsystem überhaupt realisierbar sind. Zum Beispiel ist die dreidimensionale Darstellung auf einem normalen Monitor nur mit Einbußen machbar, da richtige Tiefeninformationen fehlen und viele Daten verdeckt werden können. Weiterhin muss sich die Abbildung stets an der konkreten Problemstellung orientieren. So bedienen Information Dashboards für Manager andere Bedürfnisse als EKG- Plots in Arztpraxen. Hieraus folgt, dass der zentrale Faktor bei der Datenabbildung der Mensch in seinem sozialen und kulturellen Umfeld ist.

Während in westlichen Kulturkreisen gerne Missstände farblich rot markiert werden, würde ein chinesischer Manager diese farbliche Kodierung unter Umständen als Hinweis auf Glück und Reichtum auffassen. Ein Beispiel aus der Informatik ist die Beschreibung der Programmstruktur mit Hilfe von UML¹⁰-Diagrammen. Diese Diagramme sind für den Entwickler eine sehr verständliche Möglichkeit für Übersichtlichkeit und Vergegenwärtigung einer Struktur zu sorgen. Soll ein Manager Informationen zum Aufbau der Anwendung erhalten, werden UML-Diagramme allerdings eher für Verwirrung sorgen. Noch viel allgemeiner sind Probleme, die entstehen, wenn bei der Abbildung der Daten grundsätzliche Eigenschaften der menschlichen Wahrnehmung vernachlässigt werden, wie eine Rot-Grün-Sehschwäche (sh. auch Kap. 2.3.3, S. 20).

Deussen (2002) schlägt zur Abbildung von Daten eine Vorgehensweise in drei Schritten vor. Zuerst sollen die Eigenschaften der Daten und die Ziele der Visualisierung bestimmt werden. Daraufhin werden unterschiedliche Visualisierungsformen auf ihre Eignung geprüft. Es folgt eine Anpassung der gewählten Repräsentationen an die vorgesehenen Ziele. Kommt es zur Auswahl der passenden Visualisierung für die Datentypen, so können mögliche Kodierungen, geordnet nach ihrer Eignung laut Abbildung 4, gewählt werden.

Abbildung, Abstraktion und Interpretation von Daten erfordern eine semantische Unterscheidung zweier Formen von Daten. Nach Ware (2004) sind dies *Entitäten* und *Relationen*. Entitäten sind die zu visualisierenden Objekte, während Relationen die Strukturen und die Verhältnisse zwischen den Entitäten darstellen. Dabei müssen möglicherweise die Relationen erst durch die Visualisierung als Muster identifiziert werden.

Die Daten, die durch Entitäten oder Relationen ausgedrückt werden, können zudem *Attribute* besitzen (Anzahl, Höhe, Farbe, Anteile), wobei für den jeweiligen Fall sicherzustellen ist, ob ein Attribut nicht möglicherweise mehr Eigenschaften einer Entität besitzt. Beispielsweise kann ein Anleger Anteile an einem Unternehmen besitzen. Dann können diese Anteile als Attribut des Anlegers gesehen werden. Andererseits können die Anteile auch Entitäten sein, die mit dem Anleger in Relation stehen.

¹⁰Unified Modeling Language

Nominal	Ordinal	Quantitativ
Position	Position	Position
Länge	Dichte	Farbton
Winkel	Sättigung	Dichte
Neigung	Farbton	Sättigung

Abbildung 4: Datentypen und ihre möglichen Kodierungen nach Eignung geordnet (nach Deussen, 2002)

Die Abbildung von Daten und die spätere Interpretation der Grafik und der daraus resultierenden Informationen können nur dann geschehen, wenn die Daten einen Abstraktionsprozess durchlaufen. Hierbei soll Abstraktion nach Strothotte (1998) als „Prozess, durch den ein Auszug eines Informationsraumes gefiltert wird um die Wichtigkeit der Eigenschaften des zugrundeliegenden Modells für den Dialogkontext und das Ziel der Visualisierung widerzuspiegeln“ verstanden werden. Durch Reduktion der Details verringert sich auch die kognitive Belastung des Nutzers. Zusammen mit den Ausführungen von Noik (1993) ergeben sich die folgenden Stufen im Abstraktionsprozess. Während der Filterung werden Details reduziert und die Darstellung angepasst. Die *Didaktifizierung* (nach Weidenmann, 1889) sorgt für eine Anpassung der Größe, Form und Orientierung von Teilen des Modells abhängig vom Grad ihrer Details und Darstellung. Dies ist ein verzerrender Schritt. Zuletzt werden wichtige Elemente mit Hilfe visueller Attribute betont. Das können beispielsweise Farbe, Schriftart oder Linienstärke sein.

Interpretation von Grafiken Informationsgrafiken bestehen aus vielen verschiedenen Elementen, deren Funktionen aufgrund von Konventionen interpretiert werden können. Bertin (1967) führt für die Interpretation von Grafiken und Datenmengen die sogenannten *drei Ebenen des Lesens* ein. Er nennt sie *elementar*, *intermediär* und *umfassend*. Diese Einteilung bezieht sich auf den Umfang der jeweiligen Interpretation. Dabei kann entweder ein einzelnes Datenelement betrachtet werden, eine Gruppe von Elementen oder die Gesamtheit der Elemente. Zur Interpretation der Elemente müssen die angestrebten Aussagen in einer „Sprache“ ausdrückbar sein (Mackinlay, 1986). Unter der Annahme, dass grafische Darstellungen Sätze grafischer Sprachen sind, wobei sowohl Syntax als auch Semantik dieser grafischen

Sprachen definiert sind, ist es möglich zu bestimmen, ob eine Menge von Aussagen in einer solchen Sprache ausgedrückt werden kann.

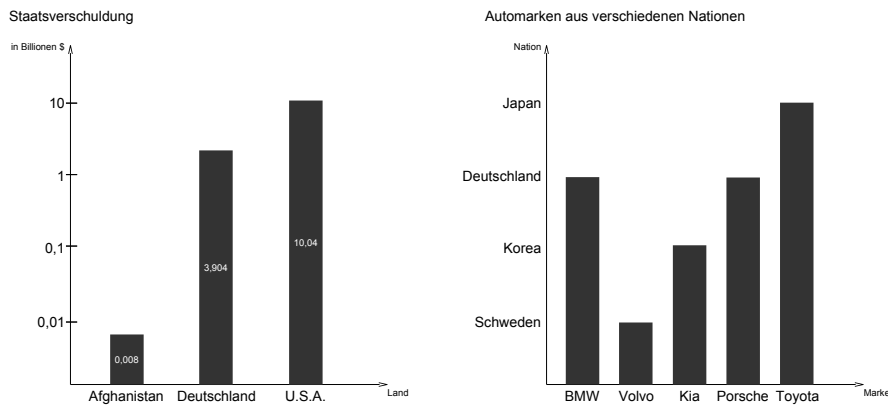


Abbildung 5: Links: Es fehlt die Information, dass der Wert für Afghanistan aus dem Jahr 2004 stammt, während für Deutschland und die U.S.A. die Verschuldung von Juni 2006 angetragen ist (Erstes Ausdruckbarkeitskriterium nach Mackinlay, 1986). Rechts: Das Balkendiagramm gibt den nominalen Nationenvariablen einen ungültigen ordinalen Anschein (Zweites Ausdruckbarkeitskriterium nach Mackinlay, 1986).

Eine Sprache muss zwei Bedingungen genügen. Zum Einen muss es einen Satz geben, der alle Aussagen in einer Menge kodiert. Zum Anderen darf dieser Satz aber auch *nur genau* die Aussagen in der Menge kodieren (Abb. 5). Wenn also die Aussage „Umsatz 2006: 1.023.320, 23 Euro“ lautet, dann ist der gesuchte Satz „Im Jahr 2006 lag der Umsatz bei 1.023.320, 23 Euro“. Der Satz „Im Jahr 2006 lag der Umsatz mit 1.023.320, 23 Euro ungewöhnlich hoch“ kodiert jedoch zusätzliche Informationen, die nicht in der Menge der Aussagen vorhanden sind und ist somit ungültig. Die linke Grafik von Abbildung 5 zeigt ein weiteres Problem der Ausdruckbarkeit bei der Kodierung von Daten auf. Dem Betrachter muss die Konvention bekannt sein, dass die horizontale Anordnung der Länder keine Auswirkung auf die Aussage der Grafik haben soll. Es wäre somit gleichwertig, eine Ordnung von $\langle U.S.A., Deutschland, Afghanistan \rangle$ aufzustellen.

Dies führt zu einer Einteilung der Datenelemente einer Grafik (im Folgenden *Variablen* genannt) gemäß ihrer inhärenter Eigenschaften in drei

Hauptgruppen nach (Card u. a., 1999). Diese Einteilung beeinflusst direkt die anwendbaren Visualisierungsformen.

Nominelle Variablen kommen aus ungeordneten Mengen wie z.B. Namen ($\{ Peter, Christoph, Adriane \}$) oder Regionen ($\{ Europa, Asien, Amerika, Australien, Afrika, Antarktis \}$) und können nur anhand von ($=, \neq$)-Relationen verglichen werden. *Ordinale* Variablen hingegen stammen aus geordneten Mengen (Tupel) wie z.B. Temperaturbereiche ($\langle kalt, kuehl, warm, heiss \rangle$), und können mit ($<, \leq$)-Relationen verglichen werden. *Quantitative* Variablen entsprechen präzisen Werten wie die Stunden des Tages ($[0, 23]$) oder einzelne Fließkommawerte (17.32, 5.16). Auf ihnen können algorithmische Operationen durchgeführt werden. Sie besitzen messbare Abstände zueinander. Quantitative Variablen können in Ordinale Variablen überführt werden. Z.B. können die beispielhaft erwähnten Zahlen (17.32, 5.16) in das Bereichs-Tupel $\langle kleinerAlsZehn, groesserAlsZehn \rangle$ transformiert werden.

	Fall 1	Fall 2
Variable 1	Wert a	Wert b
Variable 2	Wert c	Wert d
Variable 3	Wert e	Wert f

Abbildung 6: Datentabelle (Metadaten in Fettdruck)

Hilfe bei der Interpretation von Grafiken bieten sogenannte Metadaten. Metadaten sind beschreibende Informationen zu den eigentlichen Daten (Card u. a., 1999). Durch ihre Untersuchung können unter Umständen auch mögliche Visualisierungsarten zu den vorliegenden Daten gewählt werden. Stellen die Daten zum Beispiel Werte über die Zeit dar, kann eine kontinuierliche Visualisierung wie ein Liniendiagramm genutzt werden. Sind die Daten jedoch voneinander unabhängig, ist von einer solchen Form abzusehen.

Werden Metadaten geschickt in Infografiken integriert, so können sie einen erheblichen Mehrwert bedeuten. Eine Technik der wirkungsvollen Trennung von Daten und Metadaten ist das sogenannte *Layering & Separation* (Abb. 7, nach Albers, 1969). Selbst die Einteilung einer Datentabelle in Zeilen und Spalten zählt zu den Metadaten, wie in Abbildung 6 dargestellt.

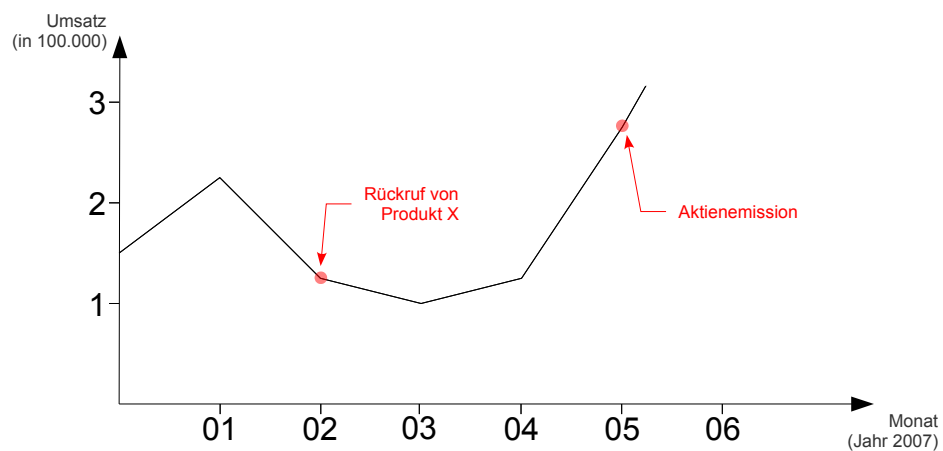


Abbildung 7: Durch deutlich vom Datenplot abgesetzte Annotationen (Layering/Separation) kann die Aussagekraft eines Diagramms erheblich verstärkt werden. (nach Albers, 1969; Tufte, 2003)

Tufte (2001) plädiert bei der Erstellung von Infografiken und der Verwendung von Metadaten auf einen großen Anteil an *Datentinte* zu achten. Datentinte sind die Teile einer Grafik, die notwendige Informationen über die zu kommunizierenden Daten beinhalten. Die vollständige Darstellung der Datentinte ist unabdingbar für das Verständnis der Grafik. Alle zusätzlichen Elemente einer Infografik sind daraufhin zu überprüfen, ob sie helfen die Aussage der Grafik zu verstehen oder sich vielleicht sogar negativ auswirken (Abb. 8). Diese zu vermeidenden Elemente bezeichnet Tufte als *Chart-Junk*.

Hierzu gehören dreidimensionale Effekte bei Kreisdiagrammen wie Spiegelungen und Schattierungen, aber auch Schraffuren, die zu einem visuellen Flackern (*Jitter*) führen. Die Reduktion der Tinte und die Konzentration auf die Datentinte kann vielfach sogar zu einer Zunahme an dargestellten Informationen führen, wie Abbildung 9 veranschaulicht. In dieser Neugestaltung eines Scatterplots wird der Median durch eine Auslassung gekennzeichnet. Die Quartilen, also die Trennungen der Werte in Viertel, werden hier von einer leichten Verschiebung des entsprechenden Achsenabschnitts repräsentiert.

Als wichtige Konvention zur visuellen Kodierung hat sich die räumliche Anordnung der abgebildeten Daten entwickelt. Im Laufe der Zeit wur-

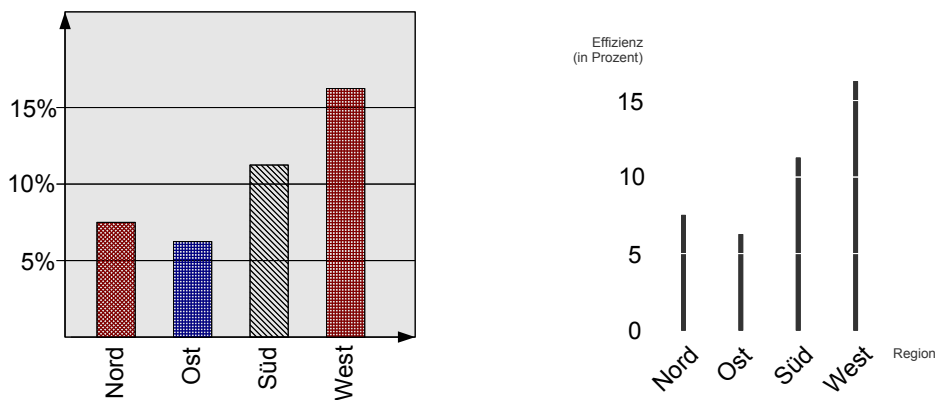


Abbildung 8: Beispiel zur Neugestaltung eines Boxplots unter Maximierung des Datentinte-Anteils (frei nach Tufte, 2001) und unter Ausnutzung der Gestaltungsgesetze (z.B. Geschlossenheit, Gute Fortsetzung – sh. Seite 28)

den verschiedene Techniken entwickelt um die Menge der kodierbaren Informationen zu maximieren. Fünf Arten (Mackinlay, 1986) sind hierbei grundsätzlich zu unterscheiden (Abb. 10).

Bei der *Composition* schafft die orthogonale Anordnung der Achsen einen metrischen Raum. Bis zu zwei Variablen können zu gut dargestellt werden, und selbst für Daten im dreidimensionalen Raum ist *Composition* möglich. Allerdings bleiben die bekannten Probleme wie Verdeckung auf zweidimensionalen Ausgabegeräten bestehen. Wird eine Achse wiederholt räumlich versetzt dargestellt, nennt sich dies *Alignment*. Dies ermöglicht unter anderem eine Vergrößerung des Datentinte-Anteils (Tufte, 2001) in Grafiken.

Eine orthogonale Fortsetzung (Faltung) der Achsen bezeichnet Mackinlay als *Folding*. Dies kann genutzt werden, wenn eine Achse für das darstellende Medium zu lang ist. In Abbildung 10 wird ein großes Computerprogramm visualisiert, wobei die Achsen die einzelnen Module darstellen und Linien den Codezeilen entsprechen. Immer wenn die Module zu umfangreich werden um komplett vertikal abgebildet zu werden, findet eine Faltung in horizontaler Richtung statt.

Die Treemaps von Ben Shneiderman (Abb. 11) sind Beispiele von *Recursion*. Der Raum zur Visualisierung der Daten wird fortlaufend weiter

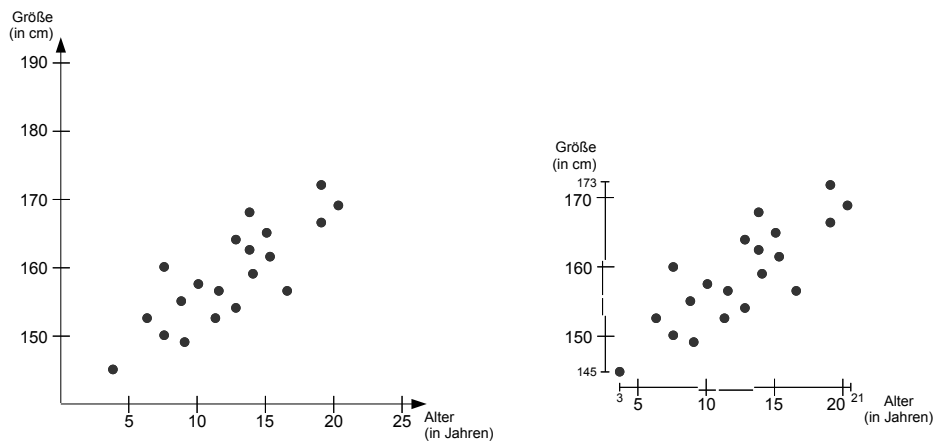


Abbildung 9: Die Entfernung der Nicht-Daten-Tinte kann auch zur Präsentation neuer Informationen genutzt werden. In diesem Scatterplot-Redesign wurden in die Achsen sowohl der jeweilige Median als auch Quartilen und Maximal- bzw. Minimalwerte integriert. (nach Tufte, 2001)

unterteilt. So kann eine hierarchische Struktur sehr kompakt dargestellt werden. Beim *Overloading* wird der zur Verfügung stehende Raum wiederverwendet um den Platz besser auszunutzen. Feiner und Beshers (1990) nutzen diese Technik bei ihren „Worlds within Worlds“ (Abb. 10).

Wahrnehmung Speziell bei der Erstellung von Grafiken zur Datenanalyse ist es wichtig zu wissen, welche Eigenheiten bei der Interpretation von Grafiken das menschliche Auge mit sich führt. Wie es sich am Beispiel der optischen Täuschungen zeigt, ist der Prozess der optischen Wahrnehmung nicht nur eine rein passive Aufnahme von Reizen. Es scheint vielmehr eine gewisse Vorverarbeitung direkt in der Netzhaut zu geschehen, die auch aufgrund einer Verknüpfung der Zapfen¹¹ beziehungsweise Stäbchen¹² möglich ist.

¹¹Die Zapfen werden zur Farbwahrnehmung benötigt. Es gibt drei Arten, die jeweils im roten, grünen oder roten Wellenlängenbereich ein Absorptionsmaximum besitzen. Sie arbeiten nur bei großer Lichtintensität

¹²Stäbchen sind extrem lichtempfindlich. Sie können aufgrund ihres Aufbaus allerdings keine Farbinformationen vermitteln. Daher ist es in der Dunkelheit nicht möglich Farben zu unterscheiden. Normales Tageslicht sättigt die Stäbchen bereits, so dass sie nahezu irrelevant für die normale Sicht sind.

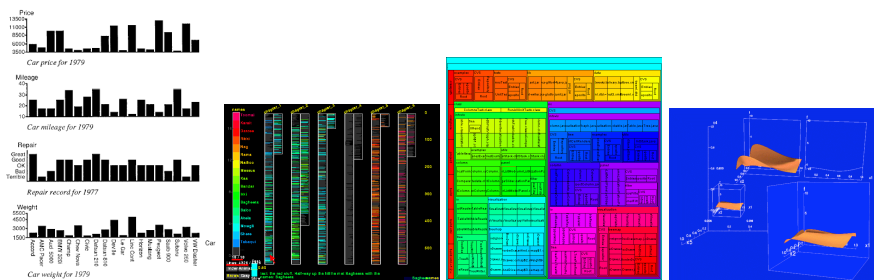


Abbildung 10: Kodierung von Information mit Hilfe von (von links nach rechts) „Alignment“, „Folding“, „Recursion“ und „Overloading“ (nach Card u. a., 1999; Mackinlay, 1986; Feiner und Beshers, 1990)

Diese subjektive Vorverarbeitung führt vor allem bei künstlich erzeugten Grafiken zu Fehlinterpretationen, die von den messbaren Merkmalen der Grafik abweichen. Solche Fehlinterpretationen können sowohl dazu führen, dass dem Nutzer wichtige Informationen entgehen, als auch dass der Nutzer Informationen identifiziert, die von den Daten nicht gestützt werden. Eine Auswahl an Wahrnehmungsfehlern soll im Folgenden vorgestellt werden.

Der Effekt des *Simultankontrastes* besteht darin, dass gleichhelle graue Flächen auf verschieden hellem Hintergrund unterschiedlich hell erscheinen (Abb. 12 und Abb. 13). Der Grund für diese falsche Wahrnehmung liegt zum großen Teil in der starken Reizung der Sehzellen durch den hellen Hintergrund, die eine dunklere Interpretation zur Folge hat¹³. Eine Abschwächung dieses Effektes wird erreicht, indem zum Beispiel durch gleichdunkle Rahmen um die grauen Flächen ein gemeinsamer Referenzhintergrund erzeugt wird. Im Falle der Datengrafiken sind also, um farb- / kontrast-kodierte Daten analysieren zu können, möglichst homogene und gleichfarbige Hintergründe zu nutzen, da sonst ein Vergleich fehlschlagen kann.

Machsche Bänder (auch *Machsche Streifen*, nach Mach, 1865) sind im Allgemeinen schwächere optische Erscheinungen als der zuvor angeführte Simultankontrast. Sie treten allerdings ebenso an den Übergängen zwischen hellen und dunklen Bereichen auf (Abb. 14). Auf der hellen Seite erscheint eine weiter erhellte Kante, auf der dunklen Seite eine abgedunkelte (Rand-

¹³In der Fachliteratur zu finden unter „Laterale Hemmung“, „Laterale Inhibition“ oder „Seitliche Hemmung“



Abbildung 11: Beispiel für eine Treemap, Quelle: InfoVis Toolkit (<http://ivtk.sourceforge.net/>)

kontrastverstärkungen). In natürlicher Umgebung finden sich Machsche Bänder zum Beispiel bei Schattenkanten.

Viel auffälliger und weit verbreitet sind Formen der *Farbfehlsichtigkeit*, die zu massiven Wahrnehmungsfehlern führen können. Das menschliche Auge besitzt drei Arten von Farbrezeptoren, sogenannte Zapfen, die jeweils zur Wahrnehmung von Rot-, Grün- oder Blautönen dienen. Fallen eine oder mehrere dieser spezialisierten Arten aus, fehlen völlig, oder sind in ihrer Funktionsweise beeinträchtigt, so spricht man von einer Farbfehlsichtigkeit.

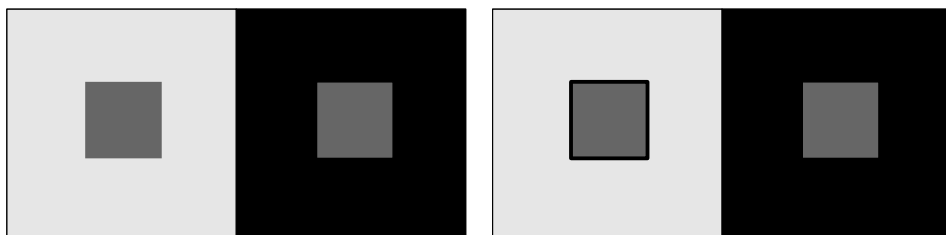


Abbildung 12: Links: Durch die helle Fläche angeregt missinterpretieren die Sehzellen das linke innere Rechteck als zu dunkel. Rechts: Der dunkle Rahmen um das linke innere Rechteck mindert den Effekt. (Simultankontrast, nach Dahm, 2006)

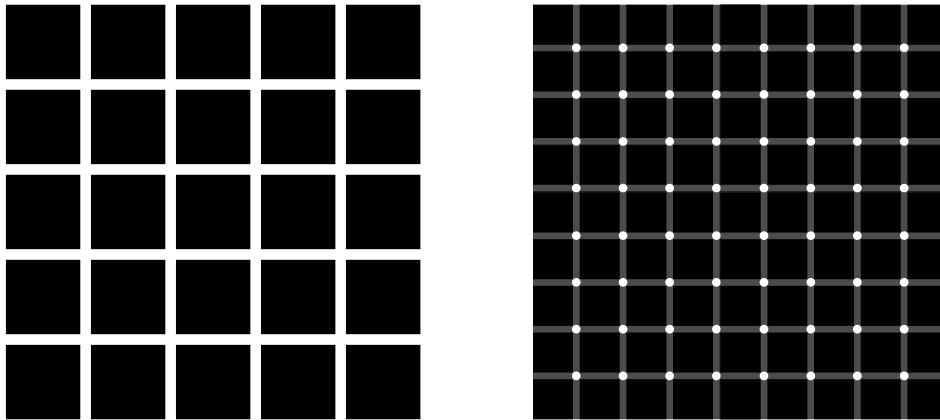


Abbildung 13: Links: Beim Hermannschen Gitter erscheinen die Kreuzungen dunkler als der Rest der vertikalen und horizontalen weißen Linien (nach Hermann, 1870). Rechts: Abwandlung nach Lingelbach und Ehrenstein Jr. (2002). Das so genannte Szintillationsgitter lässt an den weißen Kreuzungsstellen im peripheren Sichtbereich schwarze Punkte aufblitzen. Dieses Muster wurde nach Untersuchung eines nach Fourier-Analyse tiefpassgefilterten Hermann-Gitters entdeckt.

Menschen mit *Rot- Schwäche* oder *Grün- Schwäche* bereitet die Unterscheidung der Farben Braun, Gelb, Rot und Grün große Probleme.¹⁴ Weitaus seltener tritt eine *Blau- Schwäche* auf. Hierbei fällt dem Betroffenen die Unterscheidung von Rot und Orange, Blau und Grün, Grüngelb und Grau beziehungsweise Violett oder Hellgelb und Weiß schwer.

Etwa jeder zehnte Mann und jede einhundertste bis zweihundertste Frau leiden unter einem Fehler ihrer Farbwahrnehmung. Dieser bis zu zwanzigmal so hohe Wert für Männer ist auf die rezessive Vererbung der Rot- Grün- Sehschwäche auf dem 23. Chromosom (X- Chromosom) zurückzuführen. Da Frauen zwei X- Chromosomen besitzen, während das männliche Erbgut sowohl ein X- als auch ein Y- Chromosom vorweist, ist bei ihnen die Wahrscheinlichkeit höher, dass ein intaktes Chromosom die Sehschwäche deaktiviert. (Pro Patient & Gesundheit, 2005)

Für die Informationsvisualisierung ergeben sich aus dem hohen Prozentsatz der Farbfehlsichtigen in der Bevölkerung schwerwiegende Folgen. Viele

¹⁴Diese Fehlsichtigkeit wird auch als Rot- Grün- Sehschwäche oder im Extremfall (bei völligem Ausfall der entsprechenden Rezeptoren) Rot- Grün- Blindheit bezeichnet.



Abbildung 14: Bei einem kontraststarken Übergang von einem hellen Bereich in einen dunklen nimmt der Betrachter auf der hellen Seite einen aufgehellten Bereich wahr und an der dunklen Seite der Kante einen noch dunkleren. Dieser Effekt nennt sich „Machsches Band“. (nach Dahm, 2006)

handelsübliche Visualisierungswerkzeuge setzen unter anderem bei der Kennzeichnung kritischer Werte auf eine rote Farbkodierung (sogenannte Alert Icons, nach Few, 2006). Entsprechendes gilt für grün gekennzeichnete Werte, die die Anforderungen erfüllen oder übertreffen. Durchschnittlich zehn Prozent der Nutzer dieser Anwendungen oder ihrer Ausgabeprodukte werden anhand dieser Farbschlüssel keine oder falsche Folgerungen ziehen können.

Eine einfache Farbkodierung ist also nicht ausreichend. Das Problem der farbfehlsichtigen Nutzer kann allerdings umgangen werden, indem zum Beispiel die Informationen zusätzlich auf Basis der Sättigung übermittelt werden, die auch beeinträchtigte Nutzer unterscheiden können (Abb. 15). Weiterhin kann eine Kodierung durch einfache Formen erwogen werden (Abb. 16). (Few, 2006)

Die Wahl der Farben ist ein durchaus kritischer Aspekt der Informationsvisualisierung. Um durch Farbeinsatz positive Effekte erzielen zu können sind grundlegende Richtlinien zu beachten (Ware, 2004). Die Unterscheidbarkeit zwischen Vorder- und Hintergrund darf nicht nur durch verschiedene Farben gegeben sein, sondern auch durch einen starken Helligkeitskontrast.

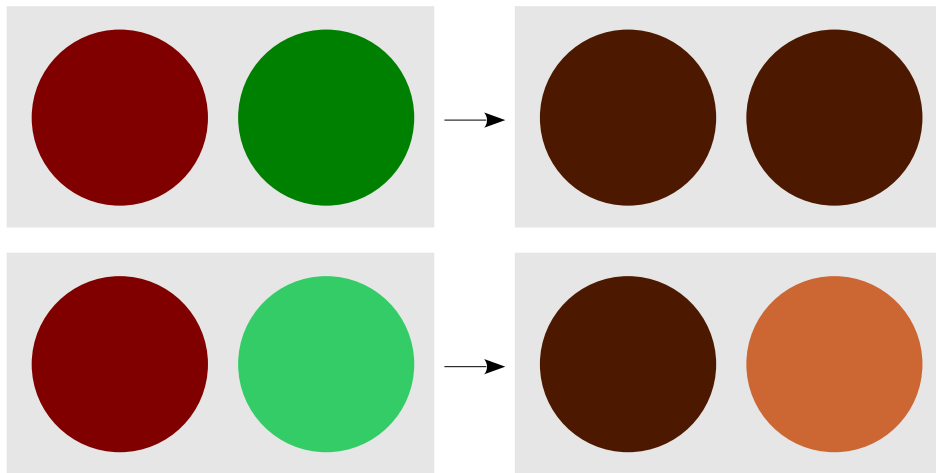


Abbildung 15: Oben: Die voll gesättigten roten und grünen Flächen sind aus Sicht eines Rot-Grün-Fehlsichtigen nicht unterscheidbar
 Unten: Durch Reduzierung der Sättigung im grünen Feld wird eine deutlich bessere Unterscheidbarkeit für farbfehlsichtige Menschen erreicht. (nach Few, 2006)

Dieser Helligkeitsunterschied kann auch durch kontrastreiche Rahmen um die Symbole sichergestellt werden.

Dabei soll die Anzahl verschiedener Farben für die Kodierung von Informationen gering gehalten werden. Zu bevorzugen sind Primärfarben wie Rot, Grün, Gelb oder Blau, da sie vom menschlichen Gehirn besonders gut erkannt werden. Zugleich sind Naturfarbtöne wie Schatten-, Himmels- und Blättertöne sehr angenehm wahrzunehmen (Tufte, 2003). Schraffuren sind grundsätzlich zu vermeiden um einen ruhig Eindruck zu erhalten und falsche Interpretationen des Farbkontrastes zu umgehen.

Müssen Flächen farbig gestaltet werden, so sollen sie nur gedeckt farbig sein, während kleine Symbole gesättigtere Farben benötigen. Wichtige Ordnungen können durch monotone Änderungen der Helligkeit, der Sättigung oder des Farbtones dargestellt werden, wobei eine Kombination dieser Eigenschaften häufig vorzuziehen ist. Gesättigte Farben sind möglichst sparsam an den Stellen einzusetzen, die besondere Aufmerksamkeit erhalten sollen.

Zur Verdeutlichung von Abweichungen von einem Referenzwert erhält

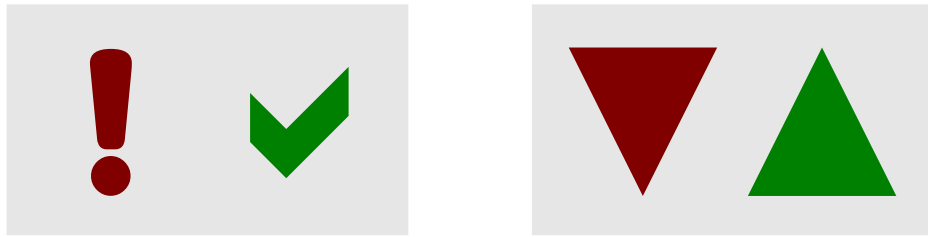


Abbildung 16: Soll die Rot / Grün- Unterscheidung zum Beispiel bei *Alert Icons* beibehalten werden, so können die Symbole trotzdem für farbfeldsichtige Menschen zugänglich gemacht werden, indem den Ikonen intuitive Formen gegeben werden, die auch ohne Farbkodierung verstanden werden. (nach Few, 2006)

der Referenzwert eine neutrale Darstellung, während stärkere Abweichungen durch jeweils stärkere Sättigung gekennzeichnet werden. Auf der technischen Seite muss darauf geachtet werden, dass das Ausgabemedium die notwendigen Farb-, Helligkeits- und Sättigungsabstufungen darstellen kann und möglichst kalibriert ist, damit die gewählten Farben eine korrekte Interpretation zulassen.

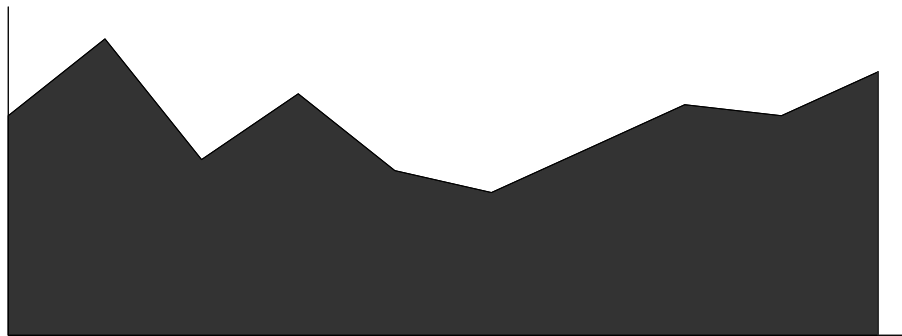


Abbildung 17: Die dunkle Schattierung unter der Linie spiegelt den uns bekannten Horizont gegen den helleren Himmel wider und hilft dem Auge, den Linienverlauf zu verfolgen. (nach Tufte, 2001)

Datengrafiken wirken jedoch nicht nur durch ihre farbliche Gestaltung. Auch ihre Proportionen und Ausmaße beeinflussen die menschliche Wahrnehmung. Grafiken sollen breiter als hoch sein, falls dies nicht ihrer Aussage widerspricht. Das menschliche Auge ist darauf spezialisiert Abweichungen und Änderungen von der Horizontlinie zu erkennen. Dies ist vermutlich

noch auf die entwicklungsgeschichtlichen Stadien zurückzuführen, als der Mensch auf der Jagd bereits von Weitem gefährliche Tiere und potenzielle Beute erkennen musste. Zudem wirkt sich die horizontale Verteilung der Augen hierfür verstärkend aus.

Für die Informationsvisualisierung kann daraus gefolgert werden, dass zum Beispiel bei Liniendiagrammen die Fläche unter der Linie dunkler gefärbt werden kann um den Verlauf herauszustellen (Abb. 17). Ein weiterer Grund für die Bevorzugung der horizontalen Ausrichtung von Informationsgrafiken ist die in westlichen Ländern vorherrschende Leserichtung von links nach rechts.

Die Wahl der Achsen sollte, wenn zutreffend, nach dem Ursache/Wirkung-Prinzip getroffen werden. Die Ursache ist an der horizontalen Achse abzutragen, um die Unterschiede in der Wirkung durch ihre Abweichung von der Horizontlinie herauszustellen. (Tufte, 2001)

Präattentivität Das Auge trägt von allen Sinnen am meisten zur menschlichen Wahrnehmung bei, und bestimmte Reize können bereits unbewusst verarbeitet werden. Durch eine auf die menschlichen Stärken angepasste Visualisierung kann der Betrachter sehr effizient Muster und Strukturen in den Daten erkennen, ohne dabei in seiner Merkleistung überfordert zu werden. Soll eine Visualisierung möglichst wirkungsvoll sein, so sollte sie sowohl Prozesse der *präattentiven* als auch der *attentiven* Wahrnehmung unterstützen. So können präattentive Merkmale genutzt werden um die Aufmerksamkeit an die Stellen zu leiten, an denen weitere attentive Untersuchungen benötigt werden. (Wright, 1995)

Präattentive Merkmale sind laut Ware (2004) Merkmale, die innerhalb von etwa zehn Millisekunden verarbeitet werden können. Im Gegensatz dazu benötigt die Verarbeitung attentiver Merkmale etwa vierzig Millisekunden (Anne Treisman, 1988). Healey wiederum gibt einen Bereich von bis zu 200-250 ms als präattentiv an. Er bezieht sich dabei auf die Zeit von 200 ms, die benötigt wird um eine Augenbewegung anzuregen.

Die Identifikation präattentiver Merkmale setzt allerdings nicht notwendigerweise eine Fokussierung voraus, so dass sie durchaus auch nahe des Randbereichs des Sichtfeldes noch zuverlässig funktioniert (Abb. 19). Innerhalb dieser Zeitspanne ist dem menschliche Sehsystem nur eine einzelne

Bewegung möglich. Eine Korrektur würde den zeitlichen Aufwand nahezu verdoppeln. Die Verarbeitung präattentiver Merkmale geschieht teilweise parallel.

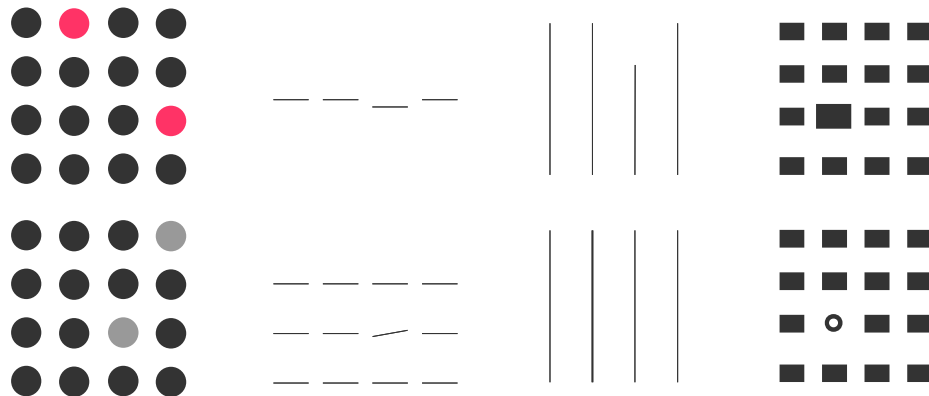


Abbildung 18: Auswahl präattentiver Merkmale nach Few (2006).
 Obere Reihe: Farbton, 2D- Position, Linienlänge, Größe
 Untere Reihe: Helligkeit, Orientierung, Linienstärke, Form

Für die Informationsvisualisierung interessante präattentive Merkmale sind Farbton, Helligkeit, Krümmung, Linienlänge und -stärke, Blinken, Orientierung, Größe, 2D- Position und (einfache) Form. Abbildung 18 zeigt ausgewählte präattentive Merkmale. Für die Kodierung quantitativer Daten sind hiervon aufgrund ihrer speziellen Eigenschaften nur Linienlänge und 2D- Position, sowie eingeschränkt Helligkeit, Größe und Linienstärke geeignet. Die restlichen Merkmale sind zwar gut unterscheidbar zu machen, wenn die Schrittweiten zwischen den diskreten Stufen groß genug gewählt werden, jedoch ist bei ihnen keine Größenordnung zu definieren, die einfach und schnell erkennbar ist.

Gestaltgesetze Der Prozess des Sehens und Erkennens läuft vor allem in zwei grundsätzlichen Phasen ab. Nach einer retinalen Vorverarbeitung der visuellen Eindrücke setzt die weitere Verarbeitung im Gehirn ein, um wichtige Teile des optischen Bildes besser zu identifizieren. (Dahm, 2006; Guski, 1996)

Bei einem Teil dieser Prozesse handelt es sich um die Erkennung „zusammengehöriger“ Elemente. Das Gehirn versucht bei der Gruppenbildung,

345324534543582149234803248024820396955091023409
234023984050293480283193243205029482348273475003
324234562193482834023943204856091329495621384338

345324534543582**1**4923480324802482039695509**1**023409
234023984050293480283**1**93243205029482348273475003
324234562**1**9348283402394320485609**1**32949562**1**384338

Abbildung 19: Der Versuch schnell die Anzahl der Einsen im oberen Text zu bestimmen wird aufgrund der komplexen Formen der Zahlen scheitern. Durch Hinzufügen eines präattentiven Merkmals kann die Anzahl (6) direkt im Überblick parallel erkannt werden. (nach Few, 2006)

eine sogenannte *Gestalt* zu finden. Die verschiedenen Arten von Gestalt wurden vor allem durch rein qualitative Beobachtungen von Max Wertheimer, Wolfgang Köhler und Kurt Koffka in den zwanziger Jahren des 20. Jahrhunderts als *Gestaltgesetze* (Organisationsprinzipien) formuliert. Die Bewegung um Koffka und die anderen Psychologen wird auch als die *Gestaltsschule der kognitiven Psychologie* bezeichnet.

Die Gestaltgesetze zeichnen sich unter anderem dafür verantwortlich, dass viele Elemente in der Informationsvisualisierung, die Tufte (2001) als *Chart Junk*¹⁵ bezeichnet, vermieden werden können, indem die Visualisierungsobjekte sich durch gemeinsame Eigenschaften optisch selbst organisieren. Im Folgenden sollen ausgewählte Gestaltgesetze vorgestellt werden, die besondere Auswirkung auf die Informationsvisualisierung haben.

Das *Gesetz der Gemeinsamen Region* (Abb. 20, rechts) führt eine visuelle Gruppenbildung auf eine Umrahmung von Objekten oder einen gemeinsamen Hintergrund zurück. Diese Eigenschaft ist vor allem bedeutsam um durch unauffällige und nicht störende Strukturierungselemente eine hohe Informationsdichte auf geringem Raum zu erhalten.

Oftmals kann aufgrund des *Gestaltgesetzes der Nähe* (Abb. 22, rechts) auf eine Umrahmung verzichtet werden, wenn genug räumlicher Abstand zwischen zu gruppierenden Objekten zur Verfügung steht.

¹⁵*Chart Junk* nennt Tufte (2001) Elemente in Grafiken, die keine Informationen tragen, so also den *Datentinte-Anteil* verringern), und die bei der Analyse störend wirken

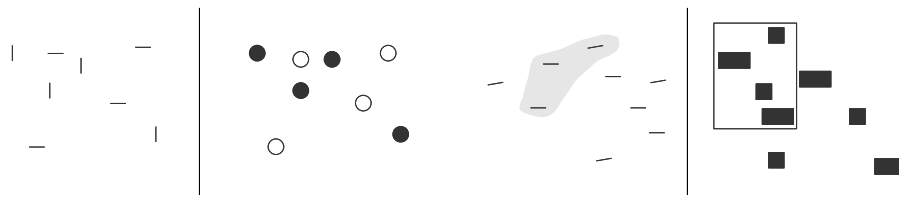


Abbildung 20: Links (Gestaltgesetz der Ähnlichkeit): Auf der linken Seite liegt eine Übereinstimmung im Winkel vor, rechts in Farbton / Helligkeit. Rechts (Gestaltgesetz der Gemeinsamen Region): Auf der linken Seite gruppiert eine gemeinsame Hintergrundfarbe die Objekte, rechts ein expliziter Rahmen.

Das *Gesetz der Ähnlichkeit* (Abb. 20, links) beschreibt die Neigung des menschlichen Gehirns zur visuellen Gruppenbildung bei Übereinstimmung von Objekten in gemeinsamen Attributen. So können verschiedene Datensätze in einem Diagramm gemeinsam dargestellt und trotzdem unterschieden werden. Ein Datensatz wird zum Beispiel mit braunen Kreisen, der andere mit grünen Rauten abgebildet.

Sollen statt diskreter Daten mehrere kontinuierliche Datensätze wie Zeitreihen mit Hilfe von Liniendiagrammen wiedergegeben werden, können Überschneidungen auftreten. Trotzdem kann das menschliche Gehirn in den meisten Fällen dem Kurvenverlauf folgen. Dies ist auf das *Gesetz der Guten Fortsetzung* (Abb. 21, rechts) zurückzuführen. Auch Unterbrechungen in Linien und Formen komplettiert das Gehirn nach dem *Gesetz der Geschlossenheit* oftmals intern, wie auch in Abbildung 21, links, zu sehen ist.

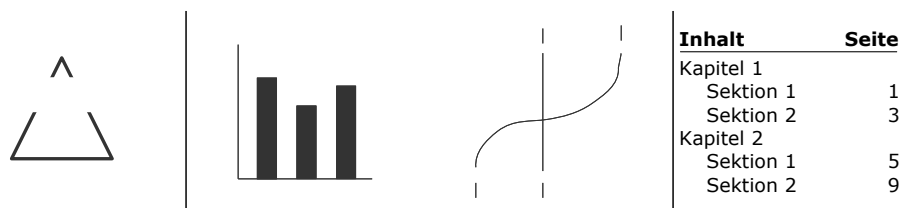


Abbildung 21: Links (Gestaltgesetz der Geschlossenheit): Ein Dreieck wird auf der linken Seite wahrgenommen. Rechts werden nur zwei Achsen zur Kennzeichnung des Diagramms benötigt. Rechts (Gestaltgesetz der Guten Fortsetzung): Auf der linken Seite werden eine gerade Linie und einer Kurve wahrgenommen. Auch die Funktion von Einrückungen ist auf dieses Gesetz zurückzuführen.

Die zwei zuvor angeführten Gestaltgesetze bauen darauf auf, dass die menschliche Wahrnehmung versucht aus einem Angebot an Interpretationen möglichst einfache Formen zu erkennen. Dies ist das *Gesetz der Guten Gestalt* (Abb. 22, links).

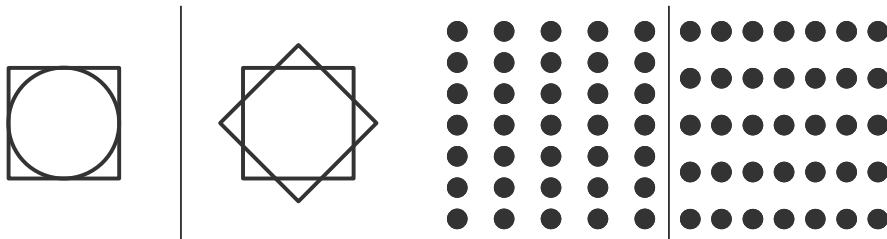


Abbildung 22: Links (Gestaltgesetz der Guten Gestalt): Auf der linken Seite nimmt der Nutzer einen Kreis und ein Quadrat wahr, rechts zwei Quadrate.
Rechts (Gestaltgesetz der Nähe): Auf der linken Seite werden Spalten erkannt, rechts Zeilen.

Die bisher vorgestellten Gestaltgesetze wirken sich je nach Situation mehr oder weniger stark auf die Wahrnehmung aus. Die in den meisten Fällen wirksamste Form der Gruppierung von Objekten ist jedoch deren Verbindung aufgrund des *Gesetzes der Verbundenheit* (Abb. 23). Ein Strich von einem Objekt zum anderen reicht oftmals aus um einen festen Bezug zwischen ihnen herzustellen.

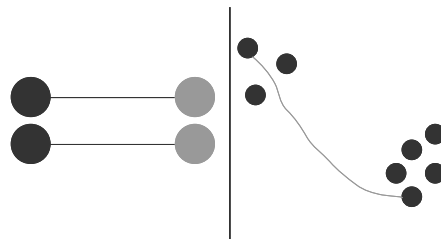


Abbildung 23: Gestaltgesetz der Verbundenheit: Auf der linken Seite wirkt die Verbundenheit stärker als die Ähnlichkeit, rechts stärker als die Nähe.

2.3.4 Formen

Tabelle und Diagramm Tabellen und Diagramme bilden bis heute das Fundament der Informationsvisualisierung. Die Eigenschaften dieser Visualisierungsformen sind bereits sehr genau untersucht worden. Sie eignen sich beide besonders zur Präsentation und Analyse quantitativer Daten. Aufgrund ihres Aufbaus zielen Tabellen und Diagramme jedoch nicht unbedingt auf die gleichen Aspekte ab. Few (2004) gibt sehr genau Auskunft über die Eigenschaften und Einsatzmöglichkeiten von Tabellen und Diagrammen, wie im Folgenden gezeigt wird.

Tabellen zeichnen sich dadurch aus, dass die präsentierten Daten in Zeilen und Spalten angeordnet und textuell kodiert sind. Die Daten werden als Wörter oder Zahlen angeführt. Obwohl Tabellenzellen oftmals durch Linien voneinander getrennt werden, ist dieses grafische Gitter nur in seltenen Fällen nützlich um die Aussagen zu unterstützen, und ihre Verwendung ist stets zu hinterfragen. Einzig die Aufteilung in Zeilen und Spalten ist essentiell. Der Haupteinsatzzweck einer Tabelle ist das Nachschlagen einzelner quantitativer Werte in Kategorien. Desweiteren können in Tabellen lokal Werte miteinander verglichen werden. Dies kann mit hoher Präzision vorgenommen werden. Zudem bieten Tabellen die Möglichkeit Informationen unterschiedlicher Metriken gemeinsam zu vermitteln.

Diagramme bieten eine größere Abstraktion der zu kommunizierenden Informationen. Sie sind „*bildliche Darstellungen quantitativer Informationen*“. Die Werte werden hier in einem durch Achsen festgelegten Raum visuell kodiert dargestellt. Um die einzelnen Werte zuordnen und vergleichen zu können, sind die Achsen beschriftet. Im Gegensatz zu Tabellen eignen sich Diagramme in erster Linie um einen Überblick über die Form, wie zum Beispiel einen Trend, der Daten zu geben. Dies führt unter Umständen zu einer geringeren Genauigkeit. Gerade da sie die Form der Daten in den Vordergrund stellen, können Diagramme dabei helfen Beziehungen zwischen mehreren quantitativen Datenquellen zu erkennen. Der Fokus bei der Nutzung von Diagrammen liegt nicht bei der exakten Ermittlung einzelner Werte, sondern bei der Erkennung von Mustern und der schnellen Interpretation großer Mengen an Informationen.

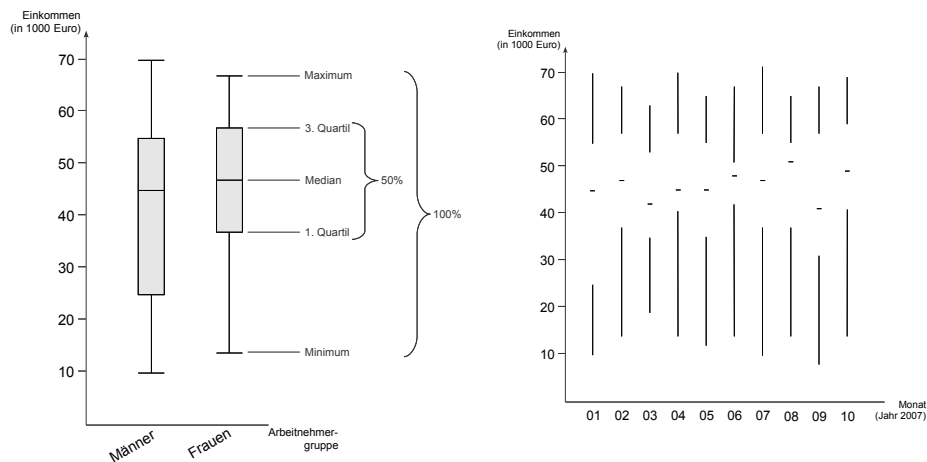


Abbildung 24: Die linke Seite zeigt den prinzipiellen Aufbau eines Box- and- Whiskers Diagramms. Rechts zu sehen ist ein Neu- Design unter Maximierung der Datentinte nach Tufte (2001).

Balkendiagramm Balkendiagramme (engl. Barcharts) kodieren quantitative Daten, indem jeweils eine senkrechte beziehungsweise waagerechte Linie beliebiger Dicke von einer der Achsen bis zum zugehörigen Wert einer anderen Achse ziehen. Aufgrund der zweidimensionalen Ausdehnung der Linien als Rechtecke, wird diese Diagrammart als Balkendiagramm bezeichnet. Die Dicke der Balken übermitteln üblicherweise keine Information, so dass prinzipiell alle Balken unterschiedliche Breite haben können.

Obwohl nur die Endpunkte der Balken die Datenwerte repräsentieren, verleihen ihre Ausmaße ihnen das nötige visuelle Gewicht um diskrete, mitunter unabhängige Daten gut differenzieren zu können. Hingegen sind Balkendiagramme ungeeignet um kategoriale Daten zu kodieren. Da das menschliche Auge immer die einzelnen Balkenlängen in Verhältnis zueinander setzt, ist bei der Wahl der Achsen unbedingt darauf zu achten, dass die Balken beim Basiswert beginnen. In den meisten Fällen wird dies die Null sein. (Few, 2004)

Box- and- Whiskers / Candlesticks Der Entdecker der Fast Fourier- Transformation (FFT) und Vorreiter auf dem Gebiet der Explorativen Datenanalyse, John Wilder Tukey, entwickelte in Tukey (1977) eine Diagrammart speziell für die Anzeige von Wertebereichen, die Box- Plots. In einem kar-

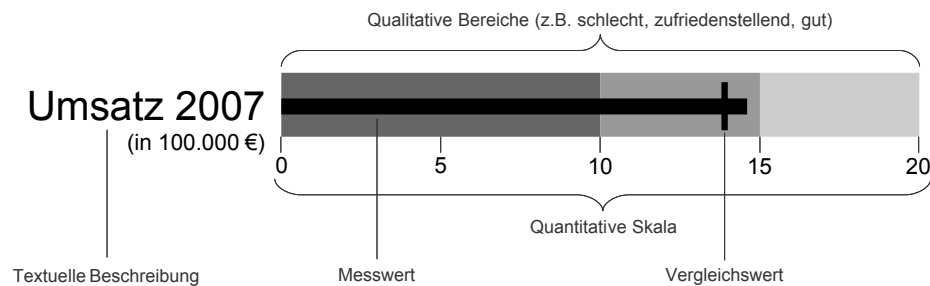


Abbildung 25: Erklärung der Elemente eines Bulletgraphs (nach Few, 2006)

thesischen Koordinatensystem markieren die obere und die untere Kante eines Rechtecks die Bereichsgrenzen eines Datensatzes. Ein waagerechter Strich kann den Mittelwert kennzeichnen.

Das sogenannte Box- and- Whiskers (deutsch: „Box und Schnurrhaare“) Diagramm führt die Informationsmaximierung auf kleinem Raum noch weiter, indem durch zwei zusätzliche Striche für den Box- Plot Maximum-, Minimum-, Mittelwert sowie oberes und unteres Quartil eingeführt werden. In Abbildung 24 wird der Aufbau nach Few (2004) erläutert.

Bulletgraph Eine besonders auch zur Erlangung einer schnellen Übersicht über Geschäftsdaten interessante Diagrammart ist der von Stephen Few entwickelte und in Few (2006) beschriebene Bulletgraph. Hierbei handelt es sich, wie in Abbildung 25 zu sehen, um ein adaptiertes Balkendiagramm, dessen Grundform jeweils aus einem einzelnen Balken besteht. Der Hintergrund dieses Balkens ist in diskrete Bereiche unterschiedlicher Helligkeit eingeteilt, die bestimmte Werte- Intervalle kodieren. Eine Skala hilft dabei, die Größenordnung der Daten einzuschätzen. Ein wichtiges Zusatzelement des Bulletgraphs ist eine Markierung in Form einer senkrecht auf dem Balken stehenden Linie, die einen Referenzwert kennzeichnet. Diese Linie bildet mit dem Balken eine präattentiv (Kap. 2.3.3, S. 27) wahrnehmbare Kreuzform, wenn der Referenzwert übertroffen wird. Wird der Wert nicht übertroffen, so sind Balken und Linien getrennt zu sehen. Auf diese Weise können auch viele Bulletgraphs gemeinsam schnell interpretiert werden. Weitere Abwandlungen des Bulletgraphs sind in Abbildung 26 zu sehen.

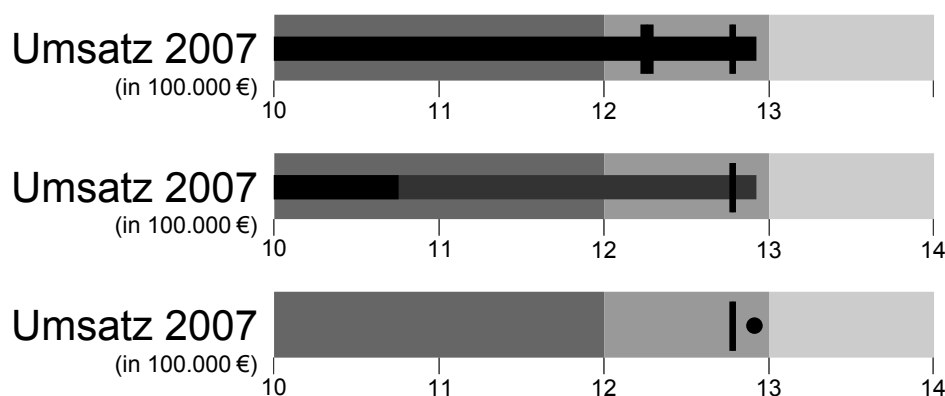


Abbildung 26: Von oben nach unten (nach Few, 2006): Vergleich mit zwei unterschiedlichen Vergleichswerten; Aktueller und voraussichtlicher Wert am Ende eines Berechnungszeitraums; Skala, die nicht null-basiert ist

Der Bulletgraph wurde von Few in erster Linie für das Gebiet der Information Dashboards entworfen, in dem Übersichtlichkeit und Effizienz höchste Priorität haben, während genau differenzierte Detail-Analysen nicht notwendigerweise möglich sein müssen. Bulletgraph konkurrieren hier vor allem mit den sogenannten Cockpits, also Anordnungen von Grafiken, die den Instrumententafeln in Kraftfahrzeugen nachempfunden sind. Die Anlehnung an eine realistische Darstellung von runden Instrumenten führt jedoch zu einem sehr schlechten Verhältnis von Information zu Platzbedarf. Genau wie die häufig in Dashboards eingesetzte Thermometer-Metapher bewirken die Cockpits zudem gerade durch den erstrebten Realismus und eventuelle Pseudo 3D- Effekte wie Spiegelungen, dass der Nutzer abgelenkt wird und seine Aufmerksamkeit stärker auf die zu kommunizierenden Informationen konzentrieren muss.

Nomogramm Nomogramme oder auch Netztafeln sind von Lalanne (1846) eingeführte Diagramme, die in erster Linie Hilfsmittel zum grafischen Rechnen sein sollen. Ihr Aufbau ermöglicht es Werte abzulesen, die von mehreren Variablen abhängig sind. Der wohl bekannteste Vorgänger der Nomogramme in Tabellenform ist die Logarithmentafel, wie in Abbildung 28 zu sehen, die dazu dient ohne Taschenrechner den Logarithmus zu einem gegebenen Numerus zu finden.

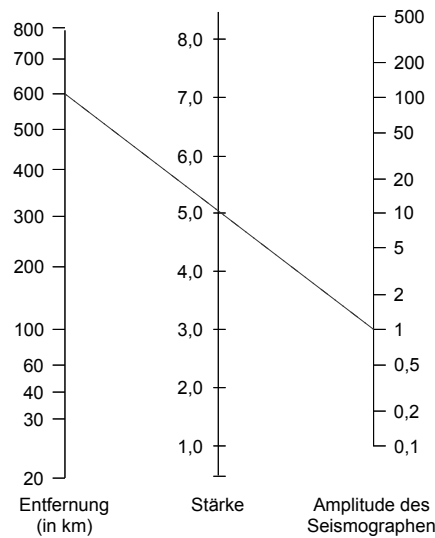


Abbildung 27: Das Nomogramm zeigt beispielhaft eine Erdbebenstärke von 5,0 für einen Seismographen- Ausschlag von 1 und der Entfernung vom Epizentrum von 600 Kilometern. (nach Arizona State University, 2004)

Ein Nomogramm besteht im Allgemeinen aus drei komplanaren Kurven, die jeweils eine Skala für eine Variable aufweisen. Die Kurven sind so anzulegen und mit den Skalen zu versehen, dass eine Gerade durch alle drei Kurven die in Beziehung stehenden Werte schneidet.

Kreisdiagramm Vor allem zur Darstellung der Hochrechnung bei Wahlen und auch in Management Dashboards, allgemein um prozentuale Angaben zu visualisieren, sind Kreisdiagramme wie in Abbildung 29 ein beliebtes Instrument. Die darzustellenden Daten werden in Kreissektoren beschrieben. Der Winkel des jeweiligen Sektors errechnet sich aus dem Verhältnis zu der Gesamtheit.

Soll zum Beispiel bei einer Bundestagswahl die Sitzverteilung visualisiert werden, und es gibt drei Parteien, die mit 30%, 50% und 20% der Sitze einziehen, so haben die Kreissektoren für die Parteien Winkel von 120°, 180° und 72°. Obwohl eine Achse nicht sichtbar ist, kann die Kreislinie – in Grad eingeteilt – als solche gesehen werden. Kreisdiagramme bieten für den Platz, den sie beanspruchen, nur wenige Informationen.

Numerus	0	1	2	3	4	5	6	7	8	9
315	83	84	86	87	89	90	91	93	94	95
316	97	98	00	01	02	04	05	06	08	09
317	50	11	12	13	15	16	17	19	20	22
318	24	26	27	28	30	31	32	34	35	37
319	38	39	41	42	43	45	46	47	49	50
320	51	53	54	56	57	58	60	61	62	64
321	65	66	68	69	70	72	73	75	76	77
322	79	80	81	83	84	85	87	88	89	91
323	92	93	95	96	97	99	00	01	03	04
324	51	05	07	08	09	11	12	13	15	17

Abbildung 28: Auszug aus einer Zehner- Logarithmentafel – Beispielsweise ist $lg(3, 170)$ in der dritten Zeile abzulesen mit 0, 5011.

Ein noch größerer Nachteil von Diagrammen, die Informationen in Flächen darstellen, ist die Tatsache, dass das menschliche Auge Unterschiede nur sehr unzuverlässig korrekt identifizieren kann. Falls doch ein Kreisdiagramm verwendet wird, so ist dafür zu sorgen, dass ein Kreissektor in der 12 Uhr Stellung, also mit einer senkrechten Kante, angelegt wird um dem Auge eine Referenz zu geben.

Liniendiagramm Die Analyse von Zeitserien und anderen kontinuierlichen Daten erfordert häufig einen Überblick über die Form und die Entwicklung, den Liniendiagramme bieten. In Liniendiagrammen werden die einzelnen Datenpunkte jeweils durch eine Linie miteinander verbunden, so dass dem Auge eine Führung und eine Art Horizont gegeben werden. Soll weiterer Fokus auf die diskreten Werte gelegt werden, so können die Datenpunkte mit Markierungen versehen werden.

Vor allem Zeitserien sollen häufig auf wiederkehrende Muster überprüft werden. Um diese Muster besser analysieren zu können eignet sich eine Spezialisierung der Liniendiagramme nach Cleveland u. a. (1978), die Cycle Charts oder Teilfolgen- Diagramme. Hierbei wird das Liniendiagramm in einzelne Unterdiagramme über bestimmte Intervalle zerlegt. Das Prinzip zeigt Abbildung 30.

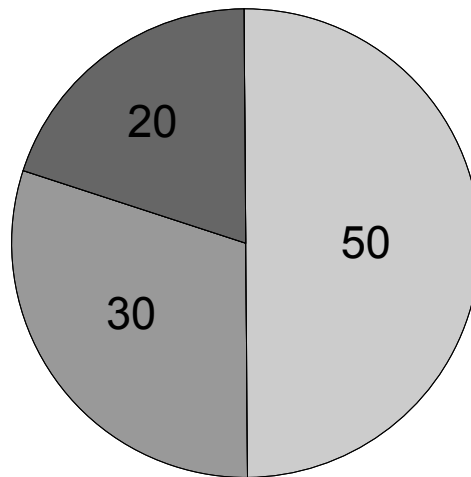


Abbildung 29: Kreisdiagramm

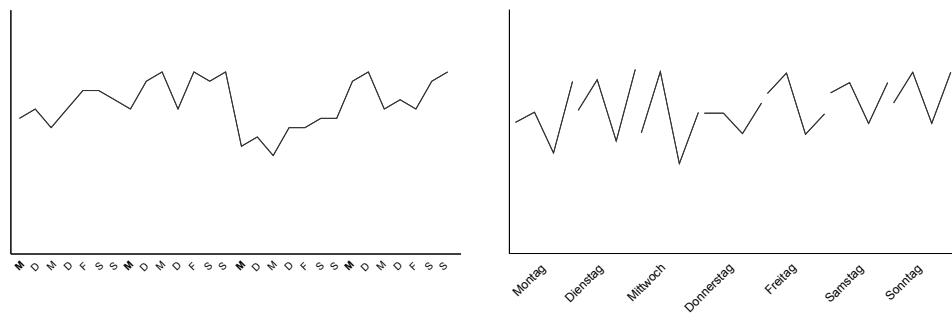
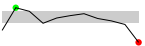


Abbildung 30: Nach einer Umstrukturierung des Liniendiagramms auf der linken Seite als Cycle-Chart wird ein Trend sichtbar.

Sparkline Edward Tufte, emeritierter Professor für Grafikdesign an der Universität Yale, entwickelte eine besondere Form der Liniendiagramme, die in einen Fließtext integriert werden kann. Sparklines, wie die Folgende , verzichten auf Achsen, da nicht die Präzision eines herkömmlichen Diagramms im Vordergrund steht, sondern der platzsparende Überblick über einen Trend. Das gerade angeführte Beispiel beinhaltet zusätzlich zu der Trendlinie noch farbliche Punktmarkierungen für den besten und den schlechtesten Wert im Beobachtungszeitraum, sowie eine hellgraue Hinterlegung des Normalbereiches.

3 Realisierung

Im Rahmen dieser Diplomarbeit sollen Hintergründe und Mechanismen der Informationsvisualisierung beleuchtet werden. Besonderer Fokus liegt hierbei auf den Möglichkeiten und Eigenarten der Nutzung der Rich Internet Application- Technologie und dem Einsatz im Umfeld von Geschäftsdaten.

Es soll eine Anwendung auf Grundlage der im vorhergehenden Kapitel recherchierten Richtlinien entwickelt werden. Dazu wurden zusammen mit Altran CIS Anforderungen und eine Priorisierung erarbeitet. Entsprechend dieser Anforderungen und einer Überprüfung der verfügbaren Technologien wurden die grafische Oberfläche der Anwendung entworfen und die erforderlichen Funktionalitäten implementiert.

3.1 Anforderungen

Im Vorfeld der Programmierung in Flex 2 wurden Anforderungen aufgestellt, die die Anwendung erfüllen soll. Zum Einen stammten diese Anforderungen aus den recherchierten Grundlagen im Rahmen dieser Diplomarbeit. Zum Anderen wurden wichtige Funktionalitäten gemeinsam mit Mitarbeitern von Altran CIS formuliert.

Das Programm soll eine Rich Internet Application sein, das heißt, eine Anwendung mit grafischer Benutzungsoberfläche, die auf einem Server vorgehalten wird und bei Bedarf im Internet- Browser des Nutzers ausgeführt wird. Die Daten sollen in dieser Version aus einer CSV- Datei importiert werden. Das genutzte Framework soll allerdings die Möglichkeiten der Datenhaltung offen halten, so dass zum Beispiel auch eine Umstellung auf eine Datenbank- Anbindung relativ einfach zu realisieren ist. Zum Testen der Applikation werden von Altran CIS möglichst realistische aber generierte Daten bereitgestellt.

Weiterhin ist es wichtig, dass die Anwendung einen hierarchisch, nach Detailstufen gegliederten, Aufbau besitzt, der von einer dashboard- ähnlichen Überblicksicht über ausführliche Ansichten der einzelnen Hauptposten bis zu dem eigentlichen Analysebildschirm reicht, dessen Anzeige dynamisch und interaktiv anpassbar ist. Dies wird sich unter Anderem in der Auswahl der angezeigten Daten und der Art der Achsen widerspiegeln.

Das Hauptaugenmerk des Analysebildschirms ist aufgrund der verwendeten Geschäftsdaten für Zeitserien zu optimieren, jedoch nicht auf diese Anwendung zu beschränken. Hinzu kommen grundlegende Filterfunktionen um den Umfang der dargestellten Daten bei Bedarf einzuschränken oder zu erweitern.

Nicht zuletzt sollen auch Hypothesen überprüft werden, die auf der ermittelten theoretischen Basis der Informationsvisualisierung beruhen. Und den Grad der Erfüllung dieser Anforderungen bestimmen zu können, wurde ein Fragebogen erstellt, der in Anhang B zu finden ist. Die Auswertung befindet sich in Kapitel 4.

3.2 Grafische Benutzungsoberfläche

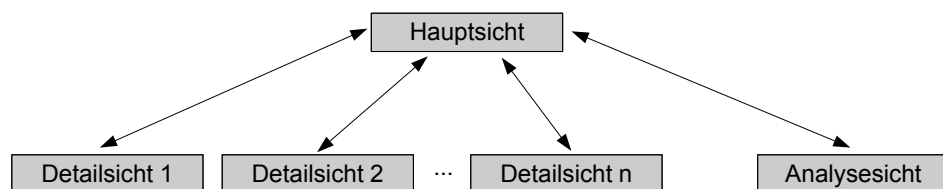


Abbildung 31: Die möglichen Navigationswege der RIA

Wie in Abbildung 31 zu sehen, sind die Navigationswege der Anwendung bewusst einfach gehalten. Es gibt außer der Hauptsicht noch jeweils eine Detailsicht für jedes Modul der Hauptsicht, sowie eine Analysesicht, die allgemein gültig ist. Die Wege zu jeder Sicht führen nur über die Hauptsicht, so dass eine Orientierung jederzeit gegeben ist.

Die Hauptsicht (Abb. 32) soll nur einen groben Überblick über Teilunternehmen und Posten geben, Trends zeigen und kritische Punkte hervorheben. Aus diesem Grund wurde eine Variante der in Kapitel 2.3.4 vorgestellten Sparklines zur Visualisierung gewählt. Die farbigen Markierungen zeigen den Start- und den Endwert, sowie den Maximum- und Minimumwert an. Um Menschen mit einer Rot-Grün Fehlsichtigkeit (Kap. 2.3.3) eine Unterscheidung der roten und grünen Markierungen zu ermöglichen, ist der verwendete Grünton nicht gesättigt. Der Effekt ist in Abbildung 15 zu sehen.

Durch die gesamte Anwendung zieht sich die bewusst unauffällig gehaltene Nutzung von Strukturierungselementen, die einer Ablenkung und

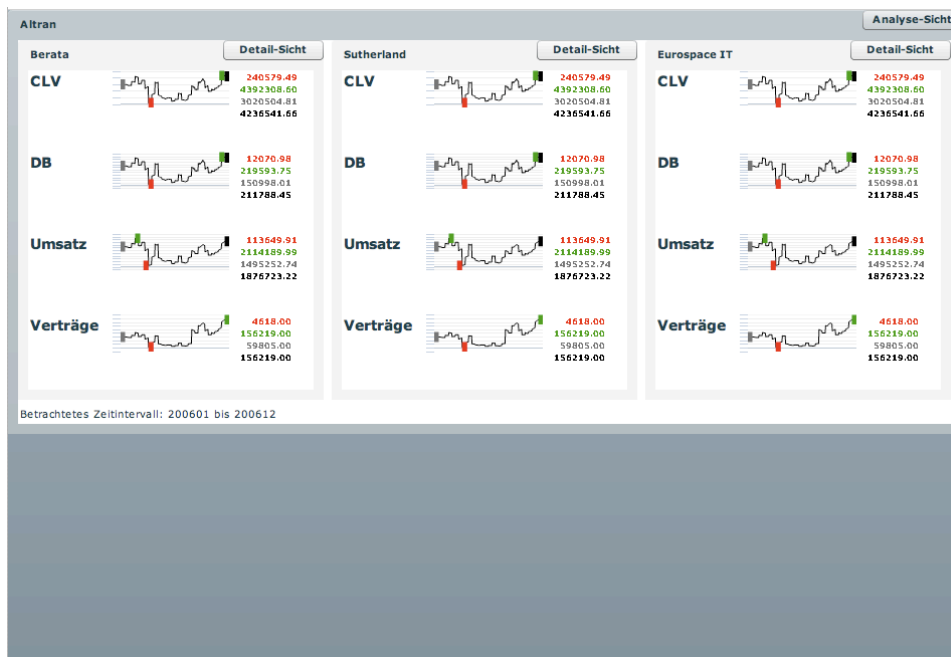


Abbildung 32: Screenshot der Haupt- Sicht (fiktive Werte)

Mehrbelastung des Nutzers entgegenwirkt. Dieses Prinzip spiegelt sich in den Rahmen der Datenmodule ebenso wider wie in den alternierenden Hintergrundfarben der Tabellen in der Detail- Sicht und den kaum sichtbaren unterstützenden horizontalen Diagrammlinien der Analyse- Sicht.

Soll schnell ein Überblick über besondere Ereignisse erhalten werden, so ist es zumeist wichtiger, negative Extrema zu bewerten, als positive, damit entsprechend schnell an kritischen Stellen gehandelt werden kann. Aus diesem Grund wird sowohl bei den Sparklines als auch in der Detail- Sicht diese Reihenfolge in der Blickrichtung von oben nach unten und von links nach rechts eingehalten. Desweiteren besitzen, wie zuvor erwähnt, die Markierungen „schlechter“ Werte eine gesättigte rote Farbe, während eine weniger gesättigte grüne Farbe „gute“ Werte anzeigt. Auch dies ist Resultat der Priorisierung negativer Extrema.

Die Detail- Sicht ist, wie in Abbildung 33 zu sehen, so aufgebaut, dass nach den Gestaltgesetzen aus Kapitel 2.3.3 zusammengehörige Objekte stets einen etwas geringeren Abstand zueinander, als zu den restlichen Elementen vorweisen. In den Tabellen zu den „Tops“ und „Flops“ wird

Eurospace IT Hauptsicht

Umsatz			Schlechtester Monat (PLZ)	
Durchschnitt			Monat	Wert
1241894.16			200601	1224820.63
Flop 3			Bester Monat (PLZ)	
PLZ-Bereich	Monat	Wert	Monat	Wert
5	200605	870732.45	200612	1259179.60
5	200609	871302.86		
5	200603	871575.03		
Top 3			Schlechteste PLZ (Monate)	
PLZ-Bereich	Monat	Wert	PLZ-Bereich	Wert
9	200612	1876723.22	5	872997.17
9	200611	1860464.83		
9	200610	1841435.05		
Customer Lifetime Value <th colspan="2">Schlechtester Monat (PLZ)</th>			Schlechtester Monat (PLZ)	
Durchschnitt			Monat	Wert
2851217.34			200602	2798523.06
Flop 3			Bester Monat (PLZ)	
PLZ-Bereich	Monat	Wert	Monat	Wert
5	200610	1933240.20	200610	2887651.96
5	200603	1949191.90		
5	200612	1955797.21		
Top 3			Schlechteste PLZ (Monate)	
PLZ-Bereich	Monat	Wert	PLZ-Bereich	Wert
9	200611	4392308.60	5	2006039.84
9	200610	4334610.11		
9	200609	4265712.15		
<td colspan="2">Beste PLZ (Monate)</td>			Beste PLZ (Monate)	
PLZ-Bereich	Monat	Wert	PLZ-Bereich	Wert
			9	4140611.77

Deckungsbeitrag			Schlechtester Monat (PLZ)	
Durchschnitt			Monat	Wert
142556.41			200602	139926.62
Flop 3			Bester Monat (PLZ)	
PLZ-Bereich	Monat	Wert	Monat	Wert
5	200610	96646.57	200610	144391.09
5	200603	97453.36		
5	200612	97762.51		
Top 3			Schlechteste PLZ (Monate)	
PLZ-Bereich	Monat	Wert	PLZ-Bereich	Wert
9	200611	219593.75	5	100296.42
9	200610	216741.74		
9	200609	213300.39		
<td colspan="2">Beste PLZ (Monate)</td>			Beste PLZ (Monate)	
PLZ-Bereich	Monat	Wert	PLZ-Bereich	Wert
			9	207017.04

Verträge			Schlechtester Monat (PLZ)	
Durchschnitt			Monat	Wert
103490			200601	102080
Flop 3			Bester Monat (PLZ)	
PLZ-Bereich	Monat	Wert	Monat	Wert
5	200612	72687	200612	104910
5	200611	72688		
5	200610	72693		
Top 3			Schlechteste PLZ (Monate)	
PLZ-Bereich	Monat	Wert	PLZ-Bereich	Wert
9	200612	156219	5	72705
9	200611	154891		
9	200610	153593		
<td colspan="2">Beste PLZ (Monate)</td>			Beste PLZ (Monate)	
PLZ-Bereich	Monat	Wert	PLZ-Bereich	Wert
			9	149259

Abbildung 33: Screenshot einer Detail- Sicht (fiktive Werte)

zudem sichergestellt, dass gemäß der Erwartung jeweils eine absteigende bzw. aufsteigende Ordnung vorliegt.

Der Fokus der Detail- Sicht liegt auf der Darstellung und der Interpretation genauer Werte. Damit dies zuverlässig und effizient geschehen kann, werden hier keine Diagramme verwendet. Einfache Tabellen sind in diesem Fall besser geeignet (Kap. 2.3.4). Bei allen Geldwerten werden programmatisch zwei Nachkommastellen erzwungen. Dies stellt sicher, dass zugehörige Stellen (10^0 , 10^1 , ...) jeweils untereinander angeordnet sind, damit die Werte schnell miteinander verglichen werden können.

Die Analyse- Sicht in Abbildung 34 ist so aufgebaut, dass die Führung an den Vorhaben des Nutzers orientiert ist. Alle Interaktionselemente, die die dargestellten Daten nicht verändern oder eindeutig einem Teil des Diagramms zugeordnet werden können, sind an den entsprechenden Stellen positioniert. Beispielsweise befinden sich die Auswahlmöglichkeiten zu den Daten der Achsen jeweils direkt an der horizontalen oder der vertikalen Achse. Filteroptionen und Möglichkeiten zur Einblendung zusätzlicher



Abbildung 34: Screenshot der Analyse- Sicht

Informationen sind in eine abgetrennte Leiste am unteren Ende des Bildschirms ausgelagert.

Die Farbigkeit der Diagramme beschränkt sich auf eine sehr dezente Palette, die an Farben aus der Natur orientiert ist. Diese Farbwahl wirkt sich wie bereits in Kapitel 2.3.3 beschrieben angenehm und beruhigend auf die menschliche Wahrnehmung aus. Die einzelnen Teildigramme können durchgängig mit Hilfe einer Legende identifiziert werden.

3.3 Adobe Flash / Flex

Die Anforderungen an die zu erstellende Anwendung führten zur Wahl der weitestverbreiteten plugin- basierten RIA- Technologie, Adobe Flash, und dem darauf basierenden Framework für RIAs, Adobe Flex 2. Flash ist ein Authoring- Werkzeug zur Erstellung interaktiver Multimedia- Inhalte für das Internet. Die primäre Zielgruppe von Flex sind Anwendungs- Entwickler datengetriebener Internet- Anwendungen. Sowohl Adobe Flash als auch Flex nutzen das SWF- Ausgabeformat.

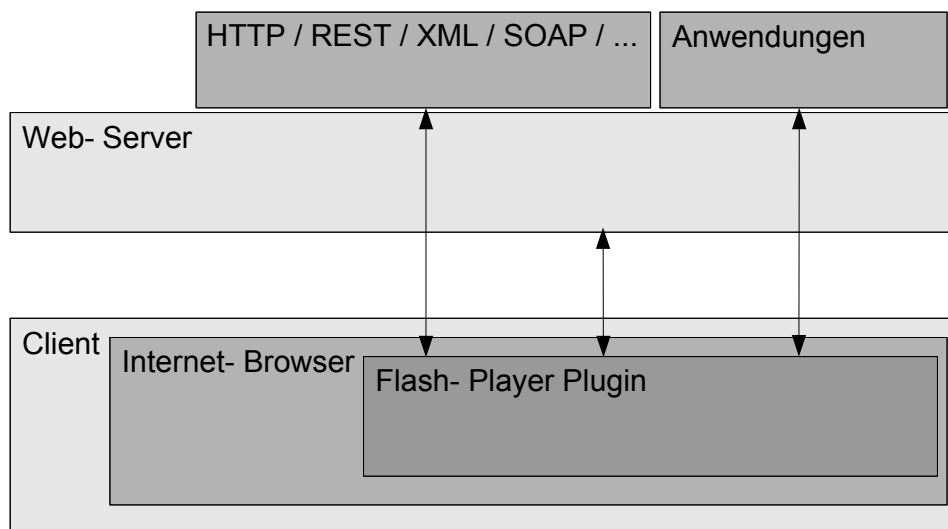


Abbildung 35: Aufbau und Kommunikation der Flex Runtime Architektur (nach Adobe Systems, Inc., 2006)

Die entwickelten Programme werden in Bytecode vorkompiliert, der im Flash- Player des Internet- Browsers von der *ActionScript Virtual Machine* ausgeführt wird. Hierzu wird der Bytecode zur Laufzeit von einem JIT¹⁶- Compiler in Maschinencode übersetzt, was eine stark verbesserte Ausführungsgeschwindigkeit bedeutet. Da die Anwendung auf dem Server als Bytecode vorliegt, setzt ein Flash- Programm nur noch das Flash- Player Plugin im Browser voraus ist ansonsten aber nicht auf bestimmte Plattformen beschränkt.

Adobe Flex in der Version 2 führt eine implizite Trennung von grafischer Benutzungsoberfläche und Programmlogik durch. Eine Anwendungsentwicklung im Sinne des MVC¹⁷- Entwurfsmusters ist somit sehr gut realisierbar. Die Quelldateien bestehen zum einen aus XML- Dateien, die die Oberfläche beschreiben und zum Anderen aus Dateien mit objektorientiertem ActionScript 3- Code.

Adobe entwickelte für den XML- Bereich einen eigenen Dialekt namens MXML. Alle MXML- Komponenten werden für die Kompilierung von Flex automatisch temporär in ActionScript umgewandelt, so dass es möglich

¹⁶*Just In Time*

¹⁷*Model View Controller*

ist, die gesamte Anwendung komplett in ActionScript zu entwickeln. Mit dem Schalter `--keep-generated-actionscript` kann der Entwickler dem Compiler mitteilen, dass die temporären Dateien nicht gelöscht werden sollen.

3.3.1 MXML

Die Hauptdatei in MXML von Flex 2- Anwendungen beinhaltet das `Application`- Tag im Standard- Namespace der Flex- Bibliotheken. Darin verschachtelt können weitere Komponenten platziert und gültige Attribute / Parameter gesetzt werden. Abbildung 36 zeigt, wie aus einer einfachen MXML- Datei eine Flex- Oberfläche entsteht.

Die vorgefertigten Flex- Komponenten, wie das `Panel` haben eine aufeinander abgestimmte Optik. Ein Entwickler kann jedoch mit Hilfe von CSS auch noch dynamisch zur Laufzeit ihr Aussehen komfortabel verändern, sowie neue Komponenten erzeugen, die er nach seinen Vorstellungen gestalten kann.

Die im Beispiel dargestellte Anwendung besitzt noch keinerlei Funktionalität. Um Interaktion und Funktionalität hinzuzufügen, muss der Programmierer die grafische Benutzungsoberfläche mittels ActionScript 3 ansprechen. Dieser Programm- Code kommuniziert mit den MXML- Komponenten.

3.3.2 ActionScript 3

ActionScript 3 (im Folgenden: AS3) ist syntaktisch gesehen eine Realisierung der ECMAScript- und ECMAScript for XML- Standards (ECMA International, 2005) und somit sehr ähnlich zu Sprachen wie JavaScript und Java.

Die Entwicklung in AS3 erfolgt auch wie in Java objektorientiert. Der Code kann entweder direkt inline über ein `Script`- Tag in die MXML- Datei eingefügt werden, oder wird über das `Script`- Tag aus externen Dateien geladen. AS3 ist gegenüber der Vorgängerversionen viel fehlerrobuster geworden, nicht zuletzt wegen der Einführung der verpflichtenden starken Bindung („strong typing“) von Konstanten, Variablen und Funktionen.

Eine lokale Fließkomma- Variable wird also mit `var variable:Number = new Number (0)` angelegt. Die Bindung in AS3 verhält sich hierbei wie der Zugriff auf Member in C++. Diese starke Bindung ermöglicht zum Einen ein sehr umfangreiches Code Hinting, zum Anderen werden viele Fehler bereits zur Compile- Zeit erkannt. Sollte sich erst während der Laufzeit der Typ einer Variablen entscheiden, so ist es möglich den Typ mit `var variable:*` als Wildcard anzugeben. So kann später die Definition zum Beispiel mit `variable = new Number ()` oder auch mit `variable = new Array ()` vorgenommen werden. Wird der Typ allerdings nicht festgelegt, so muss beim Zugriff eine Suche über eine Hash-Tabelle durchgeführt werden, was zu Geschwindigkeitseinbußen führt.

Eine der größten Änderungen zu den vorigen ActionScript- Versionen ist das neu entwickelte, auf DOM3¹⁸ basierende (Adobe Systems, Inc., 2007), Event- Modell. Interaktion zwischen dem Nutzer und Objekten sowie zwischen den Objekten untereinander werden nun nahezu vollständig über Events realisiert. Ein Event wird beispielsweise durch eine Tastatureingabe ausgelöst, aber auch, wenn eine externe XML- Datei fertig geladen wurde.

Diese Ereignisse werden überprüft, und wenn sie eintreten, können entsprechende Event- Funktionen ausgeführt werden. Dabei ist die Bindung der auslösenden Objekte an die Event- Funktionen sehr flexibel. In den Funktionen werden nicht die spezifischen Auslöser direkt behandelt, sondern ein sogenanntes Event- Objekt, das eine Schnittstelle zum eigentlichen Objekt darstellt.

Auch in der im Rahmen dieser Diplomarbeit entwickelten Anwendung spielen Events eine wichtige Rolle. Beispielfhaft soll hier ein Event vorgestellt werden, der ausgelöst wird, wenn sich der Status einer Checkbox in einer Gruppe von Checkboxes ändert. Es existiert zunächst eine von einem Behälter abgeleitete Klasse namens *MultipleCheckboxesSelection*, deren Instanz über MXML mit Checkboxes befüllt wird. Diese Checkboxes (z.B. `this.selection1`) erhalten nun eine Event- Funktion `changeHandler`, die ausgeführt wird, sobald eine Box per Maus ausgewählt wird: `this.selection1.addEventListener (MouseEvent.CLICK, changeHandler)`.

¹⁸Document Object Model, Level 3

Die zugehörige Event- Funktion sieht wie folgt aus:

```
private function changeHandler(event:MouseEvent):void
{
    if(event.target.selected)
    {
        var tmpSubStr:String = String(event.target.id);
        tmpSubStr = tmpSubStr.split("selection")[1];
        _lastSelectedIndex = Number(tmpSubStr);
    }
    updateSelectedIndices();
}
```

`event.target` bezieht sich dabei auf die jeweilige Checkbox, während `_lastSelectedIndex` eine private Variable der Klasse *MultipleCheckboxes-Selection* ist und `updateSelectedIndices` eine Methode zur Bekanntmachung der durchgeführten Änderung.

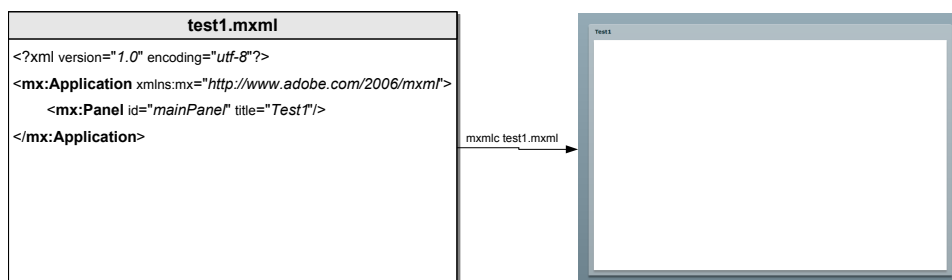


Abbildung 36: Aus einer MXML- Datei erzeugt der Flex- Compiler eine im Internet-Browser ausführbare SWF- Anwendung

3.3.3 Flex- Schnittstellen

Um eine wirkliche effiziente und umfassende Entwicklung von RIAs sicherzustellen ist das Flex- Framework dafür ausgelegt, eine hohe Interoperabilität mit anderen Technologien zu erzielen. Funktionalitäten zur Verständigung mit WebServices und XML- RPCs¹⁹ sind ebenso verfügbar,

¹⁹Extensible Markup Language- Remote Procedure Call

wie eine Integration von Flex- und AJAX- Inhalten über die Flex AJAX Bridge- Bibliotheken.

Auch die Zusammenarbeit von Flex und Java ist ohne große Hindernisse realisierbar, was vor allem in Belangen der grafischen Benutzungsoberflächen von Vorteil sein kann. Weiterhin besteht die Möglichkeit der tiefen Verlinkung („deep linking“) in eine Anwendung, so dass ein Link auf einer Web- Seite nicht nur pauschal auf die SWF- Datei, sondern auf einzelne Zustände darin, zum Beispiel Seiten einer Website, verweisen kann.

3.4 Architektur

Da die Anwendung komplett in Adobe Flex 2 entwickelt wurde, sind die Anforderungen an das Computersystem des Nutzers sehr gering. Es muss sowohl zum Laden der Anwendung eine Internetverbindung bestehen als auch ein Internet- Browser mit dem Flash Player 9- Plugin installiert sein.

Die Anwendung ist etwa 350 kB groß, so dass auch bei einer langsamen Internetanbindung nur eine kurze Ladezeit entsteht. Sollten im Laufe der weiteren Entwicklung mehr Funktionen den Umfang der Datei vergrößern, so könnte die Anwendung in Module zerlegt werden. Dadurch wird der initiale Ladevorgang kurz gehalten. Die Module könnten bei Bedarf vom Server geladen werden.

Zusätzlich zu der Anwendung werden noch eine CSS- Datei, eine XML- Datei mit Beschreibungen der Daten und eine CSV- Datei geladen. Speziell auf die XML- Datei geht auch der nächste Abschnitt noch einmal ein. Die Größe der CSV- Datei hängt von der Menge der zu analysierenden Daten ab. Zur Bereitstellung der Anwendung reicht ein Webserver. Es wird kein spezieller Applicationserver benötigt.

Die CSV- Datei wird in einem HTTP- Request angefordert. Die Standard-Sicherheitseinstellungen des Flash- Player- Plugins lassen Flash- Anwendungen in einer sogenannten *Sandbox* ausführen, wodurch der Zugriff auf externe Ressourcen eingeschränkt wird. Da alle Dateien jedoch zusammen auf einem Server liegen, treffen keine Restriktionsmaßnahmen zu.

Sobald die CSV- Datei komplett geladen ist, finden alle weiteren Operationen lokal auf dem Rechner des Nutzers statt. Die Datei wird ausgewertet, und die Daten werden in `ArrayCollections`, die einem `Array`

von assoziativen Arrays entsprechen, gespeichert. Sobald diese Vorverarbeitungsschritte beendet sind, treten nur noch geringe Verzögerungen im Programmablauf auf.

Statt der Verwendung einer CSV- Datei als Datenhalter, ist es mit wenig Aufwand möglich, die Daten zum Beispiel aus einer XML- Datei oder einer Datenbank zu beziehen. Die benötigten Funktionalitäten sind in Adobe Flex bereits integriert. Soll eine größtmögliche Unabhängigkeit von der Art der Datenquelle erreicht werden, kann die Datenholung mit Hilfe von Interfaces weiter von der Datenvorverarbeitung und der Kernanwendung getrennt werden.

3.5 Interner Programmablauf

Die Hauptdatei der für diese Diplomarbeit entwickelten Anwendung wird im SWF- Format auf einem Server vorgehalten. Wird sie angefordert, liest sie die manuell angelegte Datei *DataProvider.xml* (Abb. 37) in ein AS3 XML- Objekt namens `dataInfos` ein. Danach wird die CSS²⁰- Datei *main.css* für ein angepasstes Design der grafischen Oberflächen- Komponenten geladen. Nun geht das Programm in den Startzustand, der im `Application`- Tag festgelegt ist:

```
<mx:Application ...currentState="StartState">
```

Die Zustände der Anwendung bestehen grafisch aus jeweils einer Instanz selbst entwickelter Komponenten, die im Package `views` liegen. Da das Programm momentan drei verschiedene Sichten besitzt, gibt es in diesem Package die Komponenten `StartView` für die Haupt- Sicht, `SubsidiaryView` für die Detail- Sicht und `AnalysisView` für die Analyse- Sicht.

Nun werden die zu analysierenden Daten geladen. Dazu wird das `pfad`- Attribut von `dataInfos` ausgelesen, und die dort angegebene CSV- Datei von der ActionScript 3- Bibliothek *csvlib*²¹ in ein AS3 CSV- Objekt geschrieben. Ein Event- Handler sorgt dafür, dass, sobald die CSV- Datei fertig verarbeitet ist, die enthaltenen Daten für die Anwendung aufbereitet werden. So ist es beispielsweise notwendig, die Daten ihrer Semantik entsprechend

²⁰*Cascading Style Sheet*

²¹<http://code.google.com/p/csvlib/>

```

<DataProvider pfad="assets/companyData.csv">
  <Feld name="Monat">
    <Art> Zeit </Art>
    <Format>JJJJMM</Format>
  </Feld>
  <Feld name="Region">
    <Art>Kategorie</Art>
  </Feld>
  <Feld name="Umsatz">
    <Art>Metrik</Art>
  </Feld>
</DataProvider>

```

Abbildung 37: Beispielhafte DataProvider.xml- Datei, in die manuell semantische Informationen zu den Daten eingegeben werden

unterschiedlich zu behandeln. `dataInfos` speichert hierzu für jede Spalte der CSV- Datei die Art der Daten. Momentan sind *Zeit*, *Kategorie* und *Metrik* gültige Einträge. Die Funktion `testCorrectnessOfDataInfosXML` überprüft, ob in `dataInfos` genau die Spalten beschrieben sind, die in der CSV- Datei vorkommen. Dazu muss also eine bijektive Zuordnung bestehen. Diese Funktion ist in Abbildung 43 auf Seite 57 zu sehen.

Zudem werden mehrere `ArrayCollection`- Datenstrukturen zur späteren Verwendung mit den CSV- Daten gefüllt. Eine `ArrayCollection` ist eine Sammlung assoziativer Arrays. Sie ist der Standard- Datenquellentyp für Flex- Komponenten. Diese `ArrayCollections` ähneln sich, da sie jeweils Untermengen der Daten der CSV- Datei beinhalten.

Um Kopien beliebiger Objekte zu erstellen, wurde eine Funktion `deepCopy` programmiert, die sehr schnell echte Tiefenkopien erzeugen kann. Diese Funktion ist in Abbildung 38 zu sehen. Schließlich wird mit der `assignStartViewDataProviders`- Funktion noch die Befüllung der Elemente der Haupt- Sicht mit den zugehörigen Werten veranlasst und von der `setStartAndEndTime`- Funktion (Abb. 39) der betrachtete Zeitraum aus den Daten ermittelt.

Die Sparklines der Haupt- Sicht beziehen nun ihren Inhalt aus den gerade erzeugten Datenstrukturen und speichern sie beispielsweise in ihrer `dataProvider`- Eigenschaft. Sie bestehen grob betrachtet aus einem angepassten Liniendiagramm und Beschriftungselementen. Die Werte dieser

```

public static function deepCopy (obj:Object):*
{
    var copy:ByteArray = new ByteArray ();
    copy.writeObject    (obj);
    copy.position      = 0;
    return              (copy.readObject ());
}

```

Abbildung 38: Die Funktion `deepCopy` erzeugt eine Tiefenkopie eines beliebigen Objekts

Elemente werden dynamisch berechnet. Damit sowohl die Start- und End-Werte, als auch die besten und schlechtesten Werte der Liniendiagramme entsprechend markiert werden können, wie in Abbildung 32 auf Seite 41 zu sehen ist, musste eine spezielle Klasse namens `SparklineItemRenderer` erstellt werden, die von `ProgrammaticSkin` abgeleitet ist und das Interface `IDataRenderer` implementiert.

Das Betätigen eines Knopfes *Detail- Sicht* führt zu einem Zustandsübergang der Anwendung. Sind die grafischen Elemente der *Detail- Sicht* geladen, werden die Daten der zuvor erwähnten `ArrayCollections` in der AS3- Datei `subsidiaryView.as` verarbeitet, so dass nun zusätzlich angepasste Datenquellen vorliegen. Solche Datenquellen für Flex- Komponenten werden in der `dataProvider`- Eigenschaft der jeweiligen Komponente gespeichert.

Wird in der Haupt- Sicht nicht der Knopf *Detail- Sicht*, sondern *Analyse- Sicht* betätigt, so ändert sich der Zustand der Anwendung entsprechend. Bereits beim Start des Programms wurden alle Datenquellen für die *Analyse- Sicht* angelegt und initialisiert. Nun werden beim ersten Aufruf noch die grafischen Elemente erstellt und die jeweiligen `dataProvider` zugewiesen. Die interaktiven Oberflächenelemente erhalten Event- Funktionen, die ausgeführt werden, wenn sich der Zustand der Objekte ändert.

Im Gegensatz zu den meisten anderen Elementen der Anwendung basiert die Ansichten- Klasse `AnalysisView` nicht auf MXML, sondern ist sowohl die grafische Oberfläche als auch die Logik betreffend komplett in ActionScript 3 implementiert. Hierdurch wird eine hohe Flexibilität erreicht, was notwendig ist, da diese Sicht sehr dynamisch aufgebaut ist. Zur Laufzeit können sich große Teile ändern. So werden die Checkboxen für die Zuordnung der vertikalen Achse erst bei Bedarf aus den Überschriften der


```

private function setStartAndEndTime ():void
{
    var colsAC          :ArrayCollection;
    var timeFieldName  :String;
    var timeArr         :Array;
    var tmpXML          :XML;

    tmpXML = mx.core.Application.application.dataInfos;
    colsAC = mx.core.Application.application.csvColsCorrected;
    timeFieldName = tmpXML.Feld.(Art=="Zeit").@name;
    timeArr = new Array ();

    for (var iter0:Number = 0; iter0 < colsAC.length; iter0++)
    {
        timeArr.push (colsAC.getItemAt (iter0) [timeFieldName]);
    }

    timeArr = Misc.convertArrayToSet (timeArr).sort ();
    mx.core.Application.application.startTime = timeArr [0];
    mx.core.Application.application.endTime = timeArr
                                                [timeArr.length - 1];
}

```

Abbildung 39: Funktion zum Auslesen des Zeit- Intervalls der CSV- Daten

als „metrisch“ gekennzeichneten Spalten der CSV- Datei erstellt. Auch die Achsen und die Diagramme selber werden über Event- Handler entsprechend generiert. Die Diagramme werden auf Grundlage des ausgewählten Summe/Durchschnitt- Radiobuttons am unteren Rand des Bildschirms aggregiert und bei Bedarf nach den Schlüsseln am linken unteren Bildschirmrand (hier „Region“ und „Tochterunternehmen“) gefiltert. Für diese Filterung wird einer der Schlüssel angewählt und der *Setzen-* Knopf betätigt.

Daraufhin öffnet der Flex- Popup- Manager ein modales Popup- Fenster welches mit den entsprechenden Daten befüllt wird. Abbildung 40 zeigt diesen Zustand. Zur Befüllung der dort dargestellten ComboBox holt die Komponente die Werte aus den entsprechenden Spalten der CSV- Datei und konvertiert sie mit der statischen Helferfunktion `Misc.convertArrayCollectionToSet`, die in Abbildung 42 zu sehen ist, in eine Menge. Das bedeutet, dass eventuelle redundante Daten entfernt werden. Dieser Schritt ist notwendig, da die CSV- Datei Permutationen über einige Spalten enthält. Wird das Fenster wieder geschlossen, werden die Diagramme neu gezeichnet.



Abbildung 40: Modales Popup- Fenster zur Filterung des Analyse- Diagramms (fiktive Werte)

Eine Funktion, die zusätzliche Informationen einblendet, ist durch die Checkbox *Trendlinie anzeigen* wählbar. Hierbei führt die Funktion `calcLinRegression` der statischen Helferklasse `Misc` aus Abbildung 44 eine lineare Regression durch und übermittelt die benötigten Daten für eine Regressionsgerade.

Die Checkboxes *Log.- Achse* beziehungsweise *Y- Achse an Wertebereich anpassen* ändern keine Daten, sondern passen lediglich die vertikale Achse an. Erstere tauscht die Achsen- Metrik gegen eine logarithmische Einteilung, letztere zieht den Wertebereich der Diagramme auf den gesamten dargestellten vertikalen Bereich auf, so dass die Achse nicht mehr bei dem Nullwert beginnt. Beide Checkboxes ziehen keine Neuberechnungen nach sich.

Beim Überfahren der Diagramme mit dem Mauszeiger öffnet sich ein Datatip- Fenster, in dem der genaue aggregierte Wert an der Stelle sowie die Standardverteilung der für die Berechnung verwendeten Einzelwerte angezeigt werden. In Abbildung 41 ist dieses Datatip- Popup zu sehen. Es wurde hierzu eine eigene Funktion geschrieben, die die Standard- Datatip- Funk-



Abbildung 41: Screenshot der Analyse- Sicht mit geöffnetem Datatip- Fenster (fiktive Werte)

tion überschreibt. Verlässt der Mauszeiger den Bereich des Datenpunkts, so schließt sich das Datatip- Fenster automatisch wieder.

Die horizontale Achse des Diagramms kann alle nicht- metrischen Spalten der CSV- Datei darstellen. Dazu wird wieder die XML- Datenstruktur `dataInfos` ausgelesen. Die zugehörigen Spaltenüberschriften werden in einer `ComboBox`- Komponente zur Auswahl angeboten. Standardmäßig zeigt die horizontale Achse einen zeitlichen Verlauf an. Dabei führt eine veränderte Auswahl zum Austausch der horizontalen Achse und zum Neuzeichnen des Diagramms. Die Werte der `ComboBox`, die nicht ausgewählt sind, stehen für Filteraktionen in einer `DataGrid`- Komponente in der linken unteren Ecke des Bildschirms zur Verfügung.

3.6 Bedienung

Nach dem Start des Programms befindet sich die Anwendung in dem Zustand aus Abbildung 32, der Haupt- Sicht. Der Nutzer gewinnt eine

```

public static function convertArrayCollectionToSet
    (val:ArrayCollection):ArrayCollection
{
    var processedEntries:ArrayCollection = new ArrayCollection ();
    mainLoop: for (var i:Number = 0; i < val.length; i++)
    {
        for (var j:Number = 0; j < processedEntries.length; j++)
        {
            if (val.getItemAt(i) == processedEntries.getItemAt(j))
            {
                continue mainLoop;
            }
        }
        processedEntries.addItem (val.getItemAt(i));
    }
    return processedEntries;
}

```

Abbildung 42: Statische Funktion zur Umwandlung einer beliebigen `ArrayCollection` in eine `ArrayCollection`, deren Elemente jeweils nur einmalig enthalten sind

Übersicht über die Verläufe wichtiger Posten einzelner Tochterunternehmen im Betrachtungszeitraum. Durch die Sparkline- Visualisierung können sowohl Trends als auch Extremwerte besonders schnell zeitlich eingeordnet werden.

Der Nutzer besitzt nun zwei Möglichkeiten fortzufahren. Fällt ihm bei einem der Tochterunternehmen etwas Interessantes auf, dass er näher untersuchen möchte, kann er den zugehörigen Knopf *Detail- Sicht* betätigen. Möchte er eine allgemeine Analyse durchführen, so wählt er *Analyse- Sicht*.

Im ersten Fall werden dem Anwender Informationen speziell zu einem der Tochterunternehmen präsentiert, wie beispielsweise in Abbildung 33. Darunter fallen Durchschnitts-, aber vor allem vielfältige Extremwerte, so dass er hier erkennen kann, wo Handlungsbedarf besteht, aber auch wo und wann besonders gute Ergebnisse erzielt, damit positive Beispiele genau betrachtet werden können um eventuelle in schwachen Teilen helfen zu können.

Über die Haupt- Sicht kann der Nutzer in die Analyse- Sicht wechseln. Zu Beginn sieht diese wie Abbildung 34 aus. Der Nutzer muss sich nun entscheiden, welche Daten er analysieren möchte. Dazu kann er am oberen Ende der vertikalen Achse die gewünschten Datensätze mit Hilfe von

Checkboxen auswählen und an der horizontalen Achse die zu betrachtende Kategorie oder die Aggregationsart mit einer Combobox einstellen.

Über dieser Combobox befinden sich zwei Checkboxen. Das obere Bedienelement ermöglicht eine logarithmische Einstellung der vertikalen Achse. Eine Einsatzmöglichkeit hierfür ist der Vergleich von Trends in zwei Liniendiagrammen, deren Werte sehr unterschiedliche Größenordnungen haben. Dieses Problem wird in Abbildung 45 verdeutlicht. Aufgrund der Achsenskalierung des linken Diagramms wird dort unter Umständen eine falsche Schlussfolgerung gezogen.

Die untere der beiden Checkboxen veranlasst ebenso eine Änderung der vertikalen Achse. Hierbei ändert sich die Skalierung der Achse so, dass sie nicht mehr bei Null beginnt, sondern ein Intervall anzeigt, das dem Umfang der dargestellten Werte entspricht. Abbildung 46 zeigt, wie eine solche Anpassung markante Detail erst sichtbar macht. Allerdings sollte sich der Nutzer stets bewusst sein, dass die vertikale Achse angepasst ist, das sonst vor allem bei Balkendiagrammen Fehlinterpretationen zu befürchten sind.

Auch auf der unteren Werkzeugleiste stehen dem Anwender weitere Interaktionsmöglichkeiten zur Verfügung. Er kann sich dort eine Trendlinie für seine Datensätze anzeigen lassen, oder auch die Aggregation von einer Summen- zu einer Durchschnittsbildung ändern. Über die Anwahl einer Zeile der Tabelle in der linken unteren Ecke und Drücken des *Setzen...*-Knopfes kann der Nutzer die dargestellten Daten filtern. Es öffnet sich ein modales Popup-Fenster mit den gegebenen Filteroptionen.

Möchte der Anwender nun einzelne Werte der Diagramme genau erfahren, so kann er durch Überfahren der entsprechenden Stellen ein Datatip-Fenster öffnen, das ihm nicht nur den gewünschten Wert mitteilt, sondern auch die Standardabweichung der Daten an dieser Stelle.

Da der momentane Zustand der Analyse-Sicht während der Laufzeit gespeichert wird, kann der Nutzer bei Bedarf in anderen Sichten wechseln, um Informationen zu suchen. Wählt er die Analyse-Sicht erneut an, so kann er dort fortfahren, wo er seine Arbeit unterbrochen hatte.

```

private function testCorrectnessOfDataInfosXML ():void
{
    var tmpXML      :XML;
    var dpChildList :XMLList;
    var csvHeader   :ArrayCollection;
    tmpXML = (mx.core.Application.application.dataInfos as XML);
    dpChildList = tmpXML.children ();
    csvHeader = mx.core.Application.application.csvHeader;

    if (dpChildList.length() != csvHeader.length)
    {
        throw new Error
            ("Die Anzahl angegebener Felder\\
            ("+dpChildList.length()+")\\
            in der Datei 'DataProvider.xml'\\
            entspricht nicht der Anzahl\\
            an Spalten ("+csvHeader.length+")\\
            in der CSV-Datei.\\
            \nBitte korrigieren!");
    }
    else
    {
        [ArrayElementType("Boolean")]
        var testArr:Array = new Array (dpChildList.length ());

        for (var i:Number = 0; i < dpChildList.length(); i++)
        {
            for (var j:Number = 0; j < csvHeader.length; j++)
            {
                if (dpChildList [i].@name == csvHeader.getItemAt(j))
                {
                    testArr [i] = true;
                    break;
                }
            }
            if (!testArr [i])
            {
                throw new Error
                    ("Die Namen der Felder in der\\
                    Datei 'DataProvider.xml'\\
                    entsprechen nicht den\\
                    Spaltenüberschriften in\\
                    der CSV-Datei.\\
                    \nBitte korrigieren!");
                break;
            }
        }
    }
}

```

Abbildung 43: Funktion zum Vergleich der CSV- Spaltenüberschriften und der dataInfos- Attribute

```

public static function calcLinRegression
    (val:ArrayCollection):Array
{
    var xAvg:Number      = 0;
    var yAvg:Number      = 0;
    var xAvgArr:Array    = new Array ();
    var yAvgArr:Array    = new Array ();
    var slope:Number     = 0;
    var intercept:Number = 0;
    var slopeSum1:Number = 0;
    var slopeSum2:Number = 0;
    var tmpError:Number  = 0;

    for (var i:Number = 0; i < val.length; i++)
    {
        xAvgArr.push ((i + 1));
        yAvgArr.push (val.getItemAt (i) ["aggregated"]);
    }

    xAvg = Misc.average (xAvgArr, "mean");
    yAvg = Misc.average (yAvgArr, "mean");

    for (var j:Number = 0; j < val.length; j++)
    {
        tmpError = (val.getItemAt (j) ["aggregated"] - yAvg);
        slopeSum1 += ((j + 1) - xAvg) * tmpError;
        slopeSum2 += Math.pow ((j + 1) - xAvg, 2);
    }

    slope      = slopeSum1 / slopeSum2;
    intercept = yAvg - (slope * xAvg);

    return [slope, intercept, val.getItemAt (0) ["aggregatedValue"]];
}

```

Abbildung 44: Statische Funktion zur Berechnung einer Regressionsgeraden

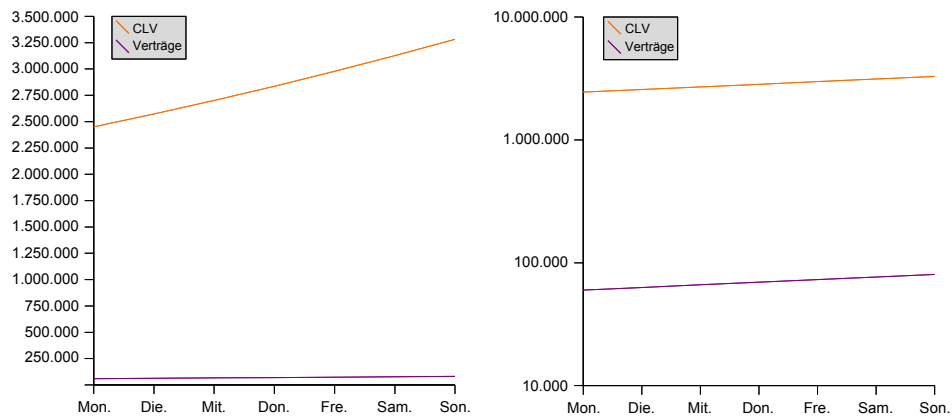


Abbildung 45: Beide Diagramme stellen die gleichen Daten dar. Jedoch wird erst durch eine logarithmische Achse deutlich, dass beide Datensätze den identischen prozentualen Änderungen über die Zeit unterliegen (hier +5%)

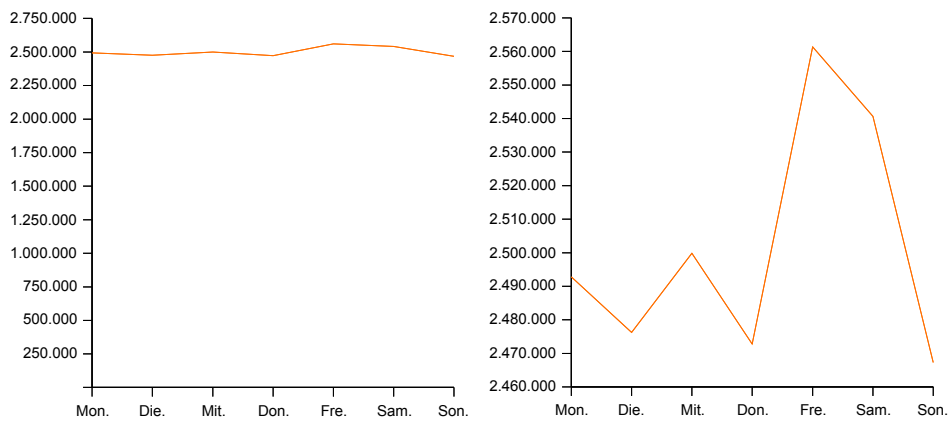


Abbildung 46: Beide Diagramme stellen die gleichen Daten dar. Jedoch kann die Anpassung des angezeigten vertikalen Bereichs wichtige Details enthüllen

4 Bewertung der Anwendung

Im Rahmen dieser Diplomarbeit wurden die Grundlagen der Informationsvisualisierung, vor allem unter dem Gesichtspunkt einer Eignung für Business Intelligence- Anwendungen, untersucht. Mit Hilfe des entwickelten Programms sollte nun eine Verbindung zu Rich Internet Application-Technologien hergestellt werden.

Auf Basis dieser Voraussetzungen wurden zusammen mit Mitarbeitern von Altran CIS Anforderungen an das Programm aufgestellt. Um zu überprüfen, in welchem Umfang die postulierten Prinzipien und die Anforderungen realisiert wurden und ob die Theorien der Informationsvisualisierung in ihrer Umsetzung die erhofften Effekte bringen, wurden die folgenden Hypothesen formuliert und mit einem Fragenbogen, der im Anhang B abgebildet ist, getestet.

Hypothese 1 Die Anwendung ist auch von Menschen mit Farbfehlsichtigkeit voll bedienbar (Fragen 0, 1)

Hypothese 2 Sparklines zeigen trotz ihrer Einfachheit zuverlässig Trends, sind gut ablesbar und lassen die zu vermittelnden Informationen ohne zusätzliche Erläuterungen gut ablesen (Fragen 2, 3, 4)

Hypothese 3 Die Navigation ist angemessen einfach (Fragen 6, 10)

Hypothese 4 Trotz der Menge an Informationen in der Detailsicht werden gesuchte Informationen schnell gefunden (Fragen 7, 8, 9)

Hypothese 5 Die Gestaltgesetze und weitere in dieser Arbeit behandelte Techniken wurden so eingesetzt, dass durch sie eine unauffällige, aber wirksame Strukturierung der grafischen Benutzungsoberfläche realisiert wurde (Fragen 5, 11, 12, 13, 14)

Der Fragebogen wurde von neun Personen ausgefüllt, wobei die Fragen 2, 8a, 13 und 14 von jeweils zwei Personen nicht beantwortet wurden und die Fragen 8b bis 12 von jeweils einer Person. Als Grund für die Nicht-Beantwortung wurde zumeist angegeben, dass die Frage nicht verstanden wurde. Eine Testperson fand den Analysebildschirm nicht, beziehungsweise wusste nicht, welche Sicht die Analysesicht war. Keiner der Probanden nutzte die angebotenen Möglichkeiten der Kontaktaufnahme, so dass eventuelle Fragen leider nicht geklärt werden konnten.

Zu Hypothese 1 Keiner der Probanden war sich einer Farbfehlsichtigkeit bewusst. Allerdings erkannten zwei der Befragten eine graue Markierung als blau. Das deutet auf eine falsche Einstellung des Monitors hin, was bei grafischen Analysen im schlimmsten Fall dazu führt, dass Zugehörigkeiten missinterpretiert werden.

Zu Hypothese 2 Der von der Sparkline angezeigte Trend wurde von allen Teilnehmern richtig identifiziert. Bis auf eine Person konnten auch alle die Position des besten Wertes erkennen. Größere Probleme machte die Bedeutung der farbigen Markierungen der Sparkline.

Ein Proband vermutete hinter dem grauen Datum (Startwert des betrachteten Zeitintervalls) den Mittelwert und konnte das schwarze Datum (Endwert des betrachteten Zeitintervalls) nicht zuordnen. Zwei Personen beantworteten Frage 2 gar nicht, da ihnen die Bedeutung der Farben vollkommen unklar schien, beziehungsweise da sie die Zahlen am rechten Rand als Skala interpretierten und dadurch verwirrt waren.

Zu Hypothese 3 Die Fragen zur Navigation wurden von allen Probanden mit „sehr einfach“ bewertet und von acht korrekt beantwortet. Ein Teilnehmer wusste nicht, welcher Bildschirm der „Analyse- Bildschirm“ ist. Von dieser Person wurden wegen dieses Problem es auch alle weiteren Fragen unbeantwortet gelassen.

Zu Hypothese 4 Bei Frage 7 gaben zwei Probanden statt des geforderten PLZ- Bereiches Monatswerte oder Geldwerte an. Sechs von neun Teilnehmern beantworteten die Teilaufgaben 8a und 8b korrekt. Falsche und nicht

abgegebene Antworten resultierten aus einem Missverständnis der Aufgabenstellungen.

Obwohl die relevanten Fragen oftmals korrekt gelöst wurden, waren die Testpersonen mit der Effizienz des Suchvorgangs eher unzufrieden. Die Bewertungen reichten von „zufrieden“ (1) über „reicht aus“ (4) bis „unzufrieden“ (3). Ein Tester gab keine Bewertung ab.

Zu Hypothese 5 Die zugehörigen Aufgaben wurden durchweg mit großem Erfolg gelöst. Die Fragen bezüglich der Analyse- Sicht wurden jedoch von einer oder zwei Personen nicht bearbeitet, da diese Sicht entweder nicht gefunden oder die Fragen nicht verstanden wurden. Bei Frage 12 wurde zudem von einem Probanden ein Wert angegeben, der nicht vorhanden war.

Bis auf die Hypothesen 1 und 4 konnten alle mit jeweils über achtzig Prozent korrekten Antworten im Umfang der Befragung bestätigt werden. Sparklines sind durchaus angemessen um schnelle Überblicke und Trends in Geschäftsdaten aufzuzeigen ohne dabei eine unnötige Komplexität aufzuweisen. Die Navigation über eine zentrale Hauptsicht ist intuitiv, muss aber für komplexer aufgebaute Anwendung zum Beispiel auf Grundlage der weit verbreiteten CUA²²- Richtlinien (I.B.M. Corp., 1993) überdacht werden.

Zudem unterstützen die Techniken zur Strukturierung der grafischen Oberflächen- und Interaktionselemente die Orientierung und Analyse effizient und effektiv, ohne dass ihr Einsatz auffällt und sie störend wirken.

Hypothese 1 konnte weder bestätigt noch widerlegt werden. Hierfür sind mehrere Gründe verantwortlich. Zum Einen ist aufgrund der begrenzten Probandenzahl die Wahrscheinlichkeit einer Person mit ausreichend starker Farbfehlsichtigkeit gering. Zum Anderen wird mit dem Fragebogen nur die weitestverbreitete Form einer Farbfehlsichtigkeit, die Rot- Grün- Sehschwäche, überprüft, während andere Arten für die erstellte Anwendung keine Relevanz besitzen.

Anhand von Hypothese 4 wird eine besondere Herausforderung der Informationsvisualisierung klar. Die Präsentation der Daten soll nicht nur effektiv sein, so dass der Nutzer der Anwendung seine Aufgaben erledigen

²²*Common User Access*

kann. Sie muss auch besonders effizient sein, damit der Nutzer zufrieden ist. Die momentanen Detailsichten der entwickelten Anwendung ermöglichen zwar, wie im Fragebogen zu sehen, durchaus, damit zusammenhängende Problem zu lösen, aber es werden bereits zu viel Informationen auf dem zur Verfügung stehenden Platz in Tabellenform dargestellt um mit befriedigendem Aufwand ausgewertet zu werden.

Zusätzlich zu den Fragen wurden auch die beiden Anmerkungsfelder intensiv von den Probanden genutzt. Wichtig für eine Rich Internet Application ist sicherlich die Ladezeit, damit die Anwendung schnell genutzt werden kann. Da die Programm- Datei nur etwa 350 kB groß ist, und weiterhin nur Textdateien geladen werden müssen, ist der Download schnell beendet. Auch die Vorverarbeitung der Daten erfolgt in kurzer Zeit. Hier kann die Rechenleistung des Client- Rechners genutzt werden. Bedienung und optische Gestaltung der grafischen Benutzungsoberfläche wurden ebenfalls als gelungen hervorgehoben.

Wie bereits aus den zugehörigen Fragen zu erahnen war, traf diese Einschätzung nicht auf die Detailsicht zu, die von zwei Probanden explizit als unübersichtlich bezeichnet wurde. Der Analysebildschirm gefiel den Probanden besser. Hier überzeugten sowohl die Möglichkeit sich die interessanten Diagramme selbst zusammenzustellen als auch das Ablesen genauer Daten durch das Führen des Mauszeigers auf die entsprechenden Stellen im Diagramm.

Die Anmerkungen, die die Probanden machten, zeigten nicht nur die bereits gelungenen Elemente der Anwendung auf. Auch wurden einige bis hierhin nur vermutete Verbesserungen ausdrücklich vorgeschlagen. Die Zahlenwerte können überschaubarer und leichter interpretiert werden, wenn sie Tausendertrennzeichen beinhalten, wodurch beispielsweise die Zahl 1955483,89 zu 1.955.483,89 würde.

Um die Anwendung für die Zielgruppe noch ansprechender zu gestalten, können Export- und Kopierfunktionalitäten für die dargestellten Werte und Diagramme implementiert werden. Desweiteren wurde vorgeschlagen, multiple Diagrammskalen beziehungsweise -achsen zu ermöglichen, um Diagramme mit weit auseinander liegenden Verläufen detaillierter anzeigen zu können. Dies führt allerdings zu der Problematik, jederzeit klar aufzuzeigen, dass die verwendeten Skalen unter Umständen stark unterschiedlich

sind, und die Werte somit nicht direkt verglichen werden dürfen. Auch muss die Zuordnung der Achsen zu den einzelnen Diagrammen prägnant zu erkennen sein. Bereits die momentane Programmversion erlaubt es aufgrund der verwendeten Daten, bis zu vier verschiedene Datensätze simultan darzustellen.

5 Zusammenfassung und Ausblick

Das vorliegende Kapitel wird eine kurze Zusammenfassung dieser Arbeit geben und Möglichkeiten aufzeigen, die im Rahmen der Diplomarbeit entwickelte Anwendung zu verbessern und zu erweitern.

5.1 Zusammenfassung

Diese Diplomarbeit befasste sich mit Problematiken der Informationsvisualisierung, vor allem im Umfeld von Geschäftsdaten und damit auch der Business Intelligence. In diesem Zusammenhang wurde speziell der Nutzen der Rich Internet Technologie durch die Umsetzung in einer Flex-Anwendung praktisch untersucht.

Unternehmen haben heute große Mengen von Daten und Informationen zu erfassen, zu verarbeiten und zu analysieren und interpretieren. Vor allem ihre Analyse wird immer komplexer, da die Daten häufig inhomogen und aus verteilten Datenquellen zu integrieren sind. Damit die Menge an vorhandenen Informationen nicht im Endeffekt zu einem Defizit an korrekt identifizierten Informationen führt, sondern effektiv für das Unternehmen genutzt werden kann, müssen genau geprüfte Analysemethoden eingesetzt werden.

Die Informationsvisualisierung kann bei der Interpretation und Analyse großer Datenmengen helfen. Dies ist zum Einen darauf zurückzuführen, dass die Informationsaufnahme des Gehirns vornehmlich auf optischem Wege zustande kommt. Zum Anderen eignet sich die grafische Darstellung und das Festhalten von Daten besonders gut als eine Art externes Gedächtnis. Die Möglichkeit, Informationen temporär außerhalb des Gehirns speichern zu können, wird wichtiger, je mehr Daten ein Mensch verarbeiten muss. Durch externe visuelle Hilfsmittel wie zum Beispiel Diagramme können Muster in den Daten gefunden, oder gefundene Muster interpretiert werden. Das menschliche Gehirn kann sich auf diese Vorgänge konzentrieren, ohne aber sich die Daten selbst merken zu müssen.

Es wurde erklärt, welche Besonderheiten bei der Abbildung von Daten zu beachten sind, damit die Visualisierung effizient und effektiv ist. Hierzu gehören sowohl die Beachtung des kulturellen Hintergrundes der Nutzer

und eventueller Wahrnehmungsfehler als auch einfach die Bestimmung der Art der Daten. Beispielsweise eignen sich Liniendiagramme aufgrund ihres kontinuierlichen Aufbaus nicht zur Abbildung diskreter Daten. Diskrete Daten sind nicht korrekt in der „Sprache“ des Liniendiagramms ausdrückbar, da die Verbindungen zwischen den Datenpunkten Informationen hinzufügen, die in der Datenmenge nicht vorhanden sind. Hinzu kamen die Prinzipien der Gestaltgesetze, die bei einer wirkungsvollen Gestaltung von Strukturierungselementen helfen können und eine Erläuterung, welche Techniken eine präattentive, teilweise parallele Wahrnehmung visueller Eigenschaften ermöglichen.

Desweiteren zeigte die Arbeit Formen von Diagrammen auf und untersuchte ihre Eignung für bestimmte Einsatzzwecke. Hier wurden ebenso bekannte und weit verbreitete Varianten wie Balken-, Linien- und Kreisdiagramm analysiert, wie insbesondere für die Visualisierung von Geschäftsdaten interessante Varianten. Hierzu gehören Sparklines, Nomogramme, Box-and-Whiskers-Plots, oder auch Bulletgraphs.

Um die Wirksamkeit der untersuchten Informationsvisualisierungs-Prinzipien zu erforschen, wurde eine Rich Internet Application entwickelt. Zusammen mit Mitarbeitern von Altran CIS wurden hierfür Anforderungen aufgestellt. Die Umsetzung erfolgte auf Basis von Adobe Flex 2. Dies ermöglichte eine weitgehend browser- und betriebsystem-unabhängige Implementierung, deren grafische Benutzungsoberfläche eher an eine Desktop-, als an eine Web-Anwendung erinnert. Auch die erreichbare Arbeitsgeschwindigkeit und die große Verbreitung führte zur Wahl dieses Frameworks.

Die Anwendung ist hierarchisch in Detailstufen gegliedert, so dass sie für unterschiedliche Nutzergruppen interessant ist. Sie besteht aus einem Übersichtsbildschirm, von dem aus genauere Sichten der einzelnen Posten oder ein allgemein zu verwendender Analysebildschirm erreicht werden können. Ein Fragebogen sollte den Grad bestimmen, zu dem die gesetzten Ziele umgesetzt wurden. Die Auswertung zeigte, dass die Anwendung als sehr positiv aufgefasst wurde. Die größten Kritikpunkte betrafen die Detailsichten und die Übersichtlichkeit aufgrund der Menge dargestellter Daten.

5.2 Ausblick

Obwohl der Fragebogen ein positives Bild der Anwendung zeichnete, zeigten die Anmerkungen der Probanden einige wichtige Verbesserungsmöglichkeiten auf. In der Programmierung wurde darauf geachtet, dass, um Zahlen gut miteinander vergleichen zu können, bei jeder Dezimalzahl zwei Nachkommastellen erzwungen werden. Weitere Klarheit kann hier, vor allem bei Zahlen im Millionenbereich durch das Hinzufügen von Tausendertrennzeichen geschaffen werden.

Allgemein gesehen steigt der Nutzwert der Anwendung, wenn die Daten nicht nur innerhalb des Programms visualisiert werden, sondern auch außerhalb zur Verfügung stehen. Hierzu könnten Exportfunktionalitäten und die Option, Werte zu kopieren, implementiert werden. Aufgrund der Sicherheitseinstellungen, die für Rich Internet Applications mit Adobe Flex gelten, sind direkte Übergaben von Werten an das Client- System nicht ohne Weiteres möglich.

Das Kopieren einzelner Werte per Tastaturkürzel ist ohne Zwischenschritt mit Flex- Bordmitteln machbar, und auch Daten aus Tabellen sind mit Hilfe serverseitiger Skriptsprachen wie PHP kommunizierbar. Sollen Diagramme exportiert werden, können sie durch die in Flash 8 eingeführte `BitmapData`- Klasse in Bitmaps umgewandelt werden. Daraufhin kann die eigentliche Bilddatei über eine Vielzahl frei verfügbarer Bibliotheken, zum Beispiel für JPG oder PDF, lokal auf dem Client- Rechner gespeichert werden.

Für das unmittelbare Drucken aus einer Flex- Anwendung kommt die `mx.printing.FlexPrintJob`- Klasse zum Einsatz. Ein Beispiel ist das Drucken einer Tabelle (in `Flex DataGrid`). Dazu soll auf Knopfdruck eine Funktion `printTable` ausgeführt werden, die wie folgt aufgebaut ist:

```
private function printTable (table:DataGrid):void
{
    var pJob:FlexPrintJob = new FlexPrintJob ();
    if (pJob.start () != true) return;
    pJob.addObject (table, FlexPrintJobScaleType.NONE);
    pJob.send ();
}
```


Die größte und eine der vielversprechendsten Weiterentwicklungen der Anwendung wäre jedoch die programmatische Erkennung der Semantik der Daten. Wie zuvor erwähnt, ist die Visualisierung immer von der Art der Daten abhängig. Es ist zu unterscheiden, ob die Daten diskret oder kontinuierlich sind, nominal oder ordinal, und ob sie zum Beispiel einen Zeitverlauf darstellen. Momentan bietet das Programm zwei Möglichkeiten zur Differenzierung:

Zum Einen werden die Einträge der eingelesenen Datenspalten mit Hilfe von regulären Ausdrücken darauf getestet, ob sie Zahlenwerte repräsentieren. Dazu werden zuerst eventuelle Anführungszeichen mit dem regulären Ausdruck `/\x22/g` anhand ihres hexadezimalen ASCII- Wertes identifiziert und mit der Funktion `replace (quotePattern, "")` entfernt. Dann wird mit `/^\d+(|[.]\d+)$/g` überprüft, ob alle Einträge der Datenspalte Zahlenwerte darstellen.

Zum Anderen wird per Hand für die aktuelle CSV- Datei eine XML- Datei namens *DataProvider.xml* angelegt, die sowohl den Pfad zu dem zugehörigen Datenhalter enthält, als auch für jede Spalte eine Beschreibung, welche Art von Daten in ihr stehen. Momentan möglich sind die Werte *Zeit*, *Kategorie* und *Metrik*. Diese XML- Datei kann beispielsweise so aufgebaut sein, wie in Abbildung 37 auf Seite 50 dargestellt.

Die entsprechende CSV- Datei sähe dann als Tabelle so aus:

Monat	Region	Umsatz
200601	Norden	2000232,22
200602	Osten	2232134,15
200603	Süden	1906252,00

Da eine komplett automatische Erkennung der Semantik nur sehr aufwändig realisierbar ist, kann kurzfristig eine alternative Lösung herangezogen werden. So ist denkbar, ein standardisiertes Datenformat einzuführen. Hierbei kämen mindestens zwei Zusatzinformationen zum Tragen. Jede Spalte der CSV- Datei erhielte einen Eintrag, ob die Daten beispielsweise metrischer, zeitlicher, kategorieller, nomineller oder anderer Art sind, sowie einen Eintrag, welche Einheit sie besitzen. Die zuvor angeführte Tabelle wäre somit wie folgt erweitert:

Monat	Region	Umsatz
<i>Zeit</i>	<i>Kategorie</i>	<i>Metrik</i>
<i>JJJJMM</i>		<i>Euro</i>
200601	Norden	2000232,22
200602	Osten	2232134,15
200603	Süden	1906252,00

Also würde der Einheiten- Eintrag in den Sonderfällen einheitenloser Spalten oder Zeit leer bleiben, beziehungsweise das vorliegende Format beschreiben.

Wie die bearbeiteten Fragebögen zeigen, stellt die im Rahmen dieser Diplomarbeit entwickelte Rich Internet Application als Proof- of- Concept einen guten ersten Schritt dar, hin zu einer verstärkten Beachtung der menschlichen Physiologie und Psychologie bei der Informationsvisualisierung und der Datenanalyse. Eine effiziente Interpretation der Geschäftsdaten wird aufgrund des ständig wachsenden Datenaufkommens immer wichtiger. Die vorliegende Arbeit versuchte, das Verständnis für die Hintergründe der Informationsvisualisierung zu schärfen und zeigte, dass jederzeit über das Internet verfügbare Analyse- Anwendungen durchaus ein Konzept der Gegenwart sind.

A Klassendiagramme

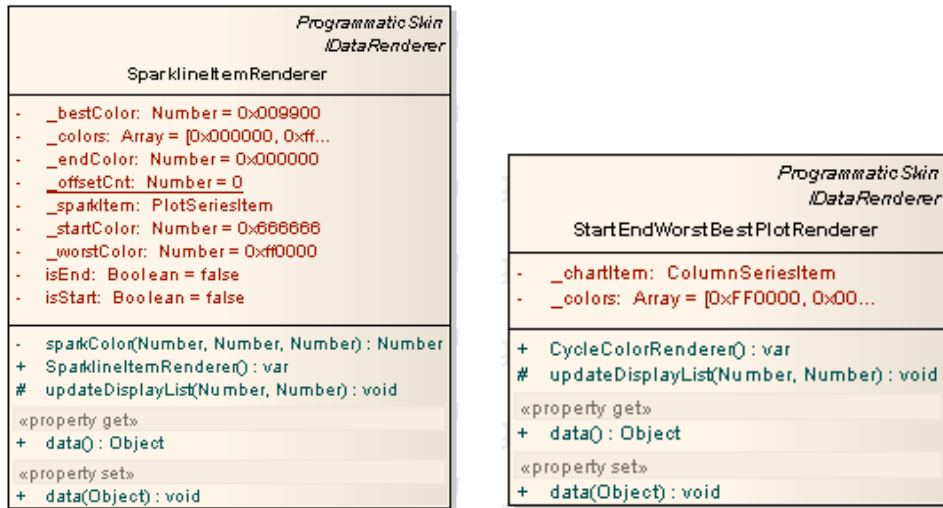


Abbildung 47: Die nötigen Anpassungen um ein Standard- Liniendiagramm in eine Sparkline zu überführen stehen in diesen beiden Klassen

B Fragenbogen

Seite 1

Vielen Dank, dass Sie sich kurz Zeit nehmen mir bei meiner Diplomarbeit zu helfen. Der vorliegende Fragebogen ist schnell bearbeitet und soll Konzepte überprüfen, die ich versucht habe in einer Dashboard- und Analyse-Anwendung umzusetzen. Das zu testende Programm ist eine sogenannte Rich Internet Application auf Basis von Adobe Flex 2. Das bedeutet, dass die etwa 350 kB große Programmdatei auf einem Server liegt, und Sie sie mit Hilfe des Flash Players in Ihrem Internet- Browser ausführen.

Voraussetzungen:

Beliebiger Internet- Browser mit installiertem Flash- Player Plugin 9.*

Bildschirmauflösung von mindestens 1024 x 768 Pixeln

Die Startseite der Flex- Applikation liegt unter der Internet- Adresse [...]. Sie benötigen eine Benutzername / Passwort Kombination. Sie lautet [...].

Bitte schauen Sie sich die Anwendung noch nicht an, sondern gehen Sie während des Ausfüllens des Fragebogens Schritt für Schritt vor! Sollten Sie sich verirren, so schließen Sie einfach das Programmfenster und klicken Sie noch einmal auf den Link der Startseite! Im Falle von Fehlermeldungen können Sie diese ignorieren und die Option „Alle schließen“ wählen, da sie keine Beeinträchtigung der Funktionalität bedeuten.

Bitte tragen Sie am Ende des Fragebogens ein, an welchen Stellen Sie besondere Probleme hatten, was Sie als besonders gelungen empfanden und wie lange Sie etwa gebraucht haben!

Sollten Sie Fragen, Probleme oder Anregungen haben, scheuen Sie sich bitte nicht, mich telefonisch zu kontaktieren! Meine Nummer lautet [...].

Seite 2

Frage 0: Wissen Sie, ob Sie eine Farb- Sehschwäche (z.B. Rot- Grün) besitzen?
Wenn ja, welche Sehschwäche haben Sie?

Zusatzinformation: Sie befinden sich in der Hauptsicht. Betrachten Sie hier für „Eurospace IT“ den Posten „DB“. Sehen Sie sich die Zahlen rechts neben dem Diagramm an.

Frage 1: Welche Farben haben die Zahlen (von oben nach unten)?

Frage 2: Ist Ihnen die Bedeutung der farbigen Markierungen der Sparklines (Liniendiagramme) klar? Wenn ja, welche Bedeutung haben sie Ihrer Meinung nach?

Frage 3: Das Diagramm stellt einen zeitlichen Verlauf dar. In welchem horizontalen Viertel befindet sich der beste Wert? (1, 2, 3 oder 4)

Frage 4: Ist der Trend im letzten Viertel eher positiv oder eher negativ?
(Zutreffendes mit „x“ ankreuzen)

Antwortmöglichkeiten: positiv, negativ

Frage 5: Aus welchem zeitlichen Bereich stammen die Daten?

Zusatzinformation: Bewegen Sie sich nun zum detaillierten Bildschirm für „Eurospace IT“.

Frage 6: Haben Sie den Weg zum nächsten Bildschirm ohne längeres Suchen/Überlegen gefunden? (Zutreffendes mit „x“ ankreuzen)

Antwortmöglichkeiten: sehr einfach, einfach, in Ordnung, schwer, nicht selbst gefunden

Frage 7: In welchem PLZ- Bereich gab es in den jeweiligen Posten die schlechtesten Werte über alle Monate?

Frage 8: Betrachten Sie den besten Umsatz für einen bestimmten Monat und einen bestimmten PLZ-Bereich! Vergleichen Sie diesen Wert nun mit der schlechtesten CLV für einen bestimmten Monat und einen bestimmten PLZ-Bereich!

a: Welcher der beiden Werte ist größer? (Zutreffendes mit „x“ ankreuzen)

b: Wie groß sind die beiden Werte?

Seite 3

Frage 9: Wie zufrieden sind Sie damit, wie schnell Sie die gewünschte Information gefunden haben? (Zutreffendes mit „x“ ankreuzen)

Antwortmöglichkeiten: sehr zufrieden, zufrieden, reicht aus, unzufrieden, nicht selbst gefunden

Frage 10: Wie kommen Sie nun zum Analyse- Bildschirm? (Schritte in Stichpunkten)

Zusatzinformation: Sie befinden sich nun in der Analyse- Sicht. Lassen Sie sich die „Verträge“ anzeigen! Statt des monatlichen Verlaufs interessiert Sie nun ein Vergleich der „Regionen“ .

Frage 11: Was müssen Sie tun, um an der horizontalen Achse statt „Monate“ die „Regionen“ abzutragen?

Frage 12: Welche Region hat die drittmeisten Verträge, und wieviele Verträge gibt es dort?

Zusatzinformation: Wechseln Sie die horizontale Achse wieder auf „Monate“! Zeigen Sie eine Trendlinie an! Die Trendlinie überlagert die Datenlinie.

Frage 13: Welche Option müssen Sie wählen, um eine genauere Sicht zu erhalten?

Zusatzinformation: Sie sehen jetzt, wieviele Verträge in allen Regionen und Tochterunternehmen in allen Monaten durchschnittlich erfasst sind.

Frage 14: Lassen Sie sich nun die Anzahl der Verträge nur für den PLZ- Bereich 6 des Tochterunternehmens Eurospace IT anzeigen! Wieviele Verträge gab es im November 2006?

Zusatzfrage 1: Haben Sie Anmerkungen zu der Anwendung unter dem Hintergrund, dass Sie sie über das Internet nutzen (Ladezeiten, Bedienung, ...)?

Zusatzfrage 2: Allgemeine Anmerkungen

Literatur

- [Adobe Systems, Inc. a] ADOBE SYSTEMS, INC.: *Adobe Flash Player Version Penetration*. – URL http://www.adobe.com/products/player_census/flashplayer/version_penetration.html. – Zugriffsdatum: 14.01.2008
- [Adobe Systems, Inc. b] ADOBE SYSTEMS, INC.: *Flash Player Penetration*. – URL http://www.adobe.com/products/player_census/flashplayer/. – Zugriffsdatum: 14.01.2008
- [Adobe Systems, Inc. 2006] ADOBE SYSTEMS, INC.: *Adobe Flex 2, Technical white paper*. Whitepaper. 2006
- [Adobe Systems, Inc. 2007] ADOBE SYSTEMS, INC.: *Programming ActionScript 3.0*. Guide. 2007
- [Albers 1969] ALBERS, Josef: *Search Versus Re-Search*. Kap. One Plus One Equals Three or More, S. 17—18, Hartford, 1969
- [Allaire 2002] ALLAIRE, Jeremy: *Macromedia Flash MX - A next-generation rich client / Macromedia*. März 2002. – Forschungsbericht
- [Anne Treisman 1988] ANNE TREISMAN, Stephen G.: *Feature analysis in early vision: Evidence from search asymmetries*. In: *Psychological Review* 95 (1988), Januar, Nr. 1, S. 15—48
- [Arizona State University 2004] ARIZONA STATE UNIVERSITY: *Project SLAM – Determining Earthquake Locations and Magnitudes*. 2004. – URL http://slam.asu.edu/activity_c.html
- [Bertin 1967] BERTIN, Jacques: *Semilogie graphique*. Gauthier-Villars, 1967
- [Card u. a. 1999] CARD, Stuart K. ; MACKINLAY, Jock D. ; SHNEIDERMAN, Ben ; CERRA, Diane D. (Hrsg.): *Readings in Information Visualization: using vision to think*. Morgan Kaufmann Publishers, Inc., 1999
- [Cleveland u. a. 1978] CLEVELAND, William S. ; DUNN, David M. ; TERPENNING, Irma: *SABL — A Resistant Seasonal Adjustment Procedure with Graphical Methods for Interpretation and Diagnosis*. In: ZELLNER, A. (Hrsg.): *Seasonal Analysis of Economic Time Series*. Washington, D.C. : U.S. Department of Commerce, Bureau of the Census, 1978, S. 201—231

- [Dahm 2006] DAHM, Markus: *Grundlagen der Mensch-Computer-Interaktion*. Pearson Studium, 2006
- [Deussen 2002] DEUSSEN, Oliver: *Datenvisualisierung*. Skript Computergraphik II, WS 02/03, TU Dresden. 2002
- [ECMA International 2005] ECMA INTERNATIONAL: *ECMAScript for XML (E4X) Specification*. Standardization. Dezember 2005
- [Feiner und Beshers 1990] FEINER, Stephen K. ; BESHES, Clifford: Worlds Within Worlds, Metaphors For Exploring N-Dimensional Virtual Worlds. In: *Proceedings of UIST'90*. Snowbird, Utah : ACM, Oktober 1990, S. 76—83
- [Few 2006] FEW, Stephen ; WHEELER, Colleen (Hrsg.): *Information Dashboard Design: The Effective Visual Communication of Data*. 1. O'Reilly Media, Inc., Januar 2006
- [Few 2004] FEW, Stephen C. ; GOLDSTEIN, Lisa (Hrsg.): *Show Me the Numbers: Designing Tables and Graphs to Enlighten*. Analytics Press, 2004
- [Fraunhofer Institut für Angewandte Informationstechnik 2006] FRAUNHOFER INSTITUT FÜR ANGEWANDTE INFORMATIONSTECHNIK: *.NET-Glossar, R*. 2006. – URL <http://zfs.informatik.rwth-aachen.de/glossar/r.aspx>
- [Grothe 2000] GROTHE, Martin: *Business Intelligence – Aus Informationen Wettbewerbsvorteile gewinnen*. 1. Addison-Wesley, 2000
- [Guski 1996] GUSKI, Rainer: *Wahrnehmen: ein Lehrbuch*. Kohlhammer, August 1996
- [Healey] HEALEY, Christopher G.: *Perception in Visualization*. – URL <http://www.csc.ncsu.edu/faculty/healey/PP/>. – Zugriffsdatum: 11.08.2007
- [Hermann 1870] HERMANN, Ludimar: Eine Erscheinung simultanen Contrastes. In: *Pflügers Archiv European Journal of Physiology* 3 (1870), Dezember, Nr. 1, S. 13—15
- [Hillringhaus und Kedzierski 2004] HILLRINGHAUS, Christoph ; KEDZIERSKI, Patrick: *Lohnt sich Business Intelligence (BI) / Fachhochschule*

Köln, Fakultät für Informations- und Kommunikationswissenschaften.
Oktober 2004. – Forschungsbericht

- [I.B.M. Corp. 1993] I.B.M. CORP.: *Object-Oriented Interface Design: IBM Common User Access Guidelines: Common User Access Development Guide*. Prentice Hall, März 1993
- [Lalanne 1846] LALANNE, Léon-Louis: Mémoire sur les tables graphiques et sur la géométrie anamorphique appliquées a diverses questions qui se rattachent a l'art de l'ingénieur. In: *Annales des Ponts et Chaussées* Bd. 11, 1846, S. 1—69
- [Lingelbach und Ehrenstein Jr. 2002] LINGELBACH, Bernd ; EHRENSTEIN JR., Walter H.: Das Hermann-Gitter und die Folgen. In: *Deutsche Optikerzeitung* 2002 (2002), Dezember, Nr. 5, S. 13—20
- [Mach 1865] MACH, Ernst: *Über die Wirkung der räumlichen Vertheilung des Lichtreizes auf die Netzhaut*. Sitzungsbericht. 1865
- [Mackinlay 1986] MACKINLAY, Jock D.: Automating the Design of Graphical Presentations of Relational Information. In: *ACM Transactions on Graphics* Bd. 5, April 1986, S. 110—141
- [Noik 1993] NOIK, Emanuel G.: Exploring large hyperdocuments: fisheye views of nested networks. In: *Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: distributed computing* Bd. 2, IBM Press, 1993, S. 661—676
- [Norman 1993] NORMAN, Donald A.: *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. Addison-Wesley, 1993
- [Playfair 1786] PLAYFAIR, William: *The Commercial and Political Atlas: Representing, by Means of Stained Copper-Plate Charts, the Progress of the Commerce, Revenues, Expenditure and Debts of England during the Whole of the Eighteenth Century*. 1786
- [Pro Patient & Gesundheit 2005] PRO PATIENT & GESUNDHEIT, Unbekannt: Farbenfehlsichtigkeit. In: *Pro Patient & Gesundheit* (2005), April, Nr. 4, S. 29—30

- [Ramsdale 2001] RAMSDALE, Chris: *The Virtual Data Warehouse: A Phase One Business Intelligence Solution*. Computer Corporation of America Whitepaper. 2001
- [Shephard und Schroeder 1995] SHEPHARD, Mark S. ; SCHROEDER, William J.: Analysis Data for Visualization. In: GALLAGHER, Richard S. (Hrsg.): *Computer visualization: graphics techniques for scientific and engineering analysis*, CRC Press, Inc., 1995, S. 61ff
- [Spence 2001] SPENCE, Robert: *Information Visualization*. Addison-Wesley, 2001
- [Strothotte 1998] STROTHOTTE, Thomas: *Computational Visualization: Graphics, Abstraction, and Interactivity*. Springer Verlag, 1998
- [Tufte 2001] TUFTE, Edward Rolf: *The Visual Display of Quantitative Information*. 2. Graphics Press, Januar 2001
- [Tufte 2003] TUFTE, Edward Rolf: *Envisioning Information*. 9. Graphics Press, August 2003
- [Tukey 1977] TUKEY, John Wilder: *Exploratory Data Analysis*. Addison-Wesley, Juni 1977
- [Ware 2004] WARE, Colin ; CERRA, Diane D. (Hrsg.): *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers, Inc., 2004
- [Weidenmann 1889] WEIDENMANN, Bernd: Informative Bilder (Was sie können, wie man sie didaktisch nutzt und wie man sie nicht verwenden sollte). In: *Pädagogik* (1889), September, S. 30—34
- [Willen 2002] WILLEN, Claudia: *Airborne Opportunities*. Januar 2002
- [Wright 1995] WRIGHT, William: Information Animation Applications in the Capital Markets. In: *Proceedings of the IEEE Information Visualization '95 Conference*, Oktober 1995, S. 19—25