

Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D - 70569 Stuttgart

Bachelorarbeit Nr. 57

**Interaktive Treemap
zur verbesserten Strategieplanung
im Unternehmenskontext**

Günther Sachs

Studiengang:	Informatik
Prüfer:	Prof. Dr. Thomas Ertl
Betreuer:	Steffen Lohmann, M. Sc.
begonnen am:	15.05.2013
beendet am:	14.11.2013
CR-Klassifikation:	E.1, H.3.5, H.5.2, H.5.3, J.1

Kurzfassung

Mit Treemaps können große hierarchische Datensätze platzsparend und übersichtlich dargestellt werden. Diese können daher unter anderem das Management eines Unternehmens bei der Strategieplanung unterstützen. Die Budgets der Abteilungen und Aufgabenbereiche werden hierbei in Form von ineinander geschachtelten Flächen dargestellt. Da sich die Kosten in einem Unternehmensbereich verändern können, müssen auch die Budgets in einer Treemap veränderbar sein. Im Rahmen dieser Bachelorarbeit wurde ein Konzept entwickelt, welches das Ändern der Budgets direkt in der Treemap ermöglicht. Abschließend wurde das Konzept zu einem Prototyp weiterentwickelt und evaluiert.

Abstract

Treemaps can be used to display large amounts of hierarchical data in a space-saving and clearly arranged manner. Therefore, Treemaps can be used to support the management of a company in strategic planning. The budgets of the departments and their responsibilities are presented in form of nested surfaces. The budget in a Treemap must be variable since the costs in a corporate area can change. In this thesis a concept was developed that enables the user to change the budgets directly in the Treemap. Finally, the concept was further developed into a prototype and then evaluated.

Inhaltsverzeichnis

1	Einleitung.....	5
2	Grundlagen	8
2.1	Treemap.....	8
2.1.1	Hierarchieebenen.....	11
2.1.2	Flächen.....	13
2.1.3	Daten.....	15
2.1.4	Treemap-Algorithmen.....	16
3	Treemaps im Unternehmenskontext	23
3.1	Einsatzgebiet.....	24
3.2	Funktions- und Designanforderungen.....	26
4	Implementierung	30
4.1	Konzept.....	30
4.2	Umsetzung.....	35
4.2.1	Datenexploration.....	37
4.2.2	Datenänderung.....	40
4.2.3	Funktionsklassifikation.....	44
5	Evaluation	46
5.1	Umfeld und Ablauf	46
5.2	Ergebnisse.....	48
5.2.1	Beobachtung	48
5.2.2	Fragebogen.....	50
5.2.3	Meinungsumfrage.....	53
5.2.4	Sitzung.....	54
6	Fazit und Ausblick	55
	Literaturverzeichnis.....	61
	Anhang.....	63

Abbildungsverzeichnis

Treemap.....	9
Vergleich Baum vs. Treemap [7].....	11
Hierarchieebenen und Farbgebung in einer Treemap.....	12
Hierarchische Liste (JSON) und Treemap-Element	16
Squarified-Algorithmus [11].....	19
Squarified-Algorithmus, Pseudocode [12].....	20
Vergleich Treemap-Algorithmen [13].....	21
Diagrammtypen.....	23
Business-Treemap.....	29
Beispiel eines Datensatzes im Unternehmensbereich.....	31
Budget-Erhöhung in einer Treemap.....	33
Budgetkürzung mit Budget-Sperren.....	34
Treemap-Implementierung, Startscreen.....	39
Treemap-Implementierung, Änderungsoptionen.....	41
Treemap-Implementierung, Datenänderungsoberfläche.....	42
Schaltflächen zum Sperren eines Budgets.....	43
Treemap-Implementierung, Funktionsklassifikation.....	45
Verbesserungsvorschlag 1.....	56
Verbesserungsvorschlag 2.....	57
Treemap Business Data Visualization [24].....	59
Business Analysis Tool Desktop [25].....	59

1 Einleitung

Seit dem Beginn des digitalen Zeitalters steigt das jährlich generierte Volumen an digitalen Daten weltweit exponentiell an. Heutzutage geht man davon aus, dass dieses über 2 Zettabyte (Stand 2013) beträgt [1]. Aktuelle Studien belegen, dass sich das Datenvolumen alle zwei Jahre verdoppelt und somit in Zukunft um ein Vielfaches steigen wird [1]. Die Ursachen dafür liegen vor allem in der zunehmenden Zahl technischer Geräte (z.B. Smartphones oder Kameras) und in der verbreiteten Nutzung immer größer werdender Netzwerke zum Datenaustausch. Auch in Unternehmen steigt das Datenvolumen rasant an. Dies führt zu neuen Herausforderungen im Bereich der effizienten Datenspeicherung und -verwaltung. Um bei dieser gewaltigen Informationsflut Daten in einer globalisierten Arbeitsumgebung effizient einzusetzen, zu verwalten und teilen zu können, werden unter anderem auch immer effizientere und übersichtlichere Visualisierungsmethoden benötigt.

Eine dieser Methoden stellt die Daten auf mehreren Ebenen hierarchisch in einer Treemap (dt. Baumkarte) dar. Diese Visualisierungsform entstand in den frühen Neunzigerjahren an der Universität Maryland durch Shneiderman und Johnson. Das Konzept wurde ursprünglich für die Darstellung der Speicherplatzverwendung auf Festplatten eines Servers erstellt. Damit konnten die Anwendungen identifiziert werden, die Speicherplatz auf einer Festplatte verschwendeten [2][3]. Seitdem wurde diese Visualisierungsmethode stetig weiterentwickelt. Somit lassen sich Treemaps heutzutage nicht nur zur Speicherverwaltung einsetzen, sondern werden auch in vielen anderen Bereichen verwendet.

Insofern eignen sich Treemaps unter anderem auch zur Darstellung von Echtzeitdaten, wie z.B. zur Darstellung von Import- und Exportwerten, zur Personalverwaltung oder zur Darstellung von Börsenkursen [4][5].

Im Rahmen dieser Bachelorarbeit soll ein Konzept zur Strategieplanung im Unternehmenskontext entworfen werden, das auf einer interaktiven Visualisierungsform basiert. Die Darstellung soll aus einem hierarchischen Datensatz modelliert werden. Zur Abbildung der hierarchischen Organisationsstrukturen wird eine Treemap als Visualisierungsform verwendet. Dem Benutzer sollen dabei interaktive Funktionalitäten zur detaillierten Exploration und Anpassung der Visualisierung bereitstehen. Das Ziel dieser Visualisierung ist die Unterstützung des Managements bei der Strategieplanung in einem Unternehmen. Die Datensätze eines Unternehmens müssen daher möglichst übersichtlich in einer Treemap dargestellt werden. Ein Anwendungsgebiet dieses Konzepts kann z.B. die Budgetverteilung an einzelne Abteilungen und Aufgabenbereiche oder die Analyse der Kostenentwicklung in einem Unternehmen sein. Die Bachelorarbeit umfasst zusätzlich die Entwicklung eines Prototyps und die Evaluation des Konzepts.

Folgende Inhalte werden in dieser Bachelorarbeit behandelt:

In **Kapitel 2** werden allgemeine Grundlagen in Bezug auf die Visualisierungsmethode 'Treemap' vermittelt, die zum besseren Verständnis der nachfolgenden Kapitel beitragen. Zusätzlich zur Definition einer Treemap werden unterschiedliche Treemap-Algorithmen und deren Einsatzmöglichkeit vorgestellt. In diesem Kontext werden sowohl die Vorteile als auch die möglichen Schwächen dieser Algorithmen erörtert.

In **Kapitel 3** wird die Verwendung von Treemaps im Unternehmenskontext angesprochen. Zudem werden die Funktionsanforderungen und Designkriterien einer Treemap vorgestellt, die für eine Verwendung in diesem Bereich benötigt werden.

In **Kapitel 4** wird das Konzept beschrieben, das zur Strategieplanung in einem Unternehmen eingesetzt werden soll. Zusätzlich dazu wird ein Prototyp vorgestellt und evaluiert, welcher unter Berücksichtigung des entworfenen Konzepts entstanden ist. Außerdem werden die verwendeten Programmiersprachen und das Toolkit beschrieben.

In **Kapitel 5** werden die Evaluationskriterien sowie die Testumgebung beschrieben und die Ergebnisse der Evaluation präsentiert.

In **Kapitel 6** dieser Bachelorarbeit werden die Ergebnisse zusammengefasst und es wird ein Einblick in den aktuellen Forschungsstand in diesem Bereich gegeben.

2 Grundlagen

Eine Treemap ist eine Visualisierungsform, welche zur Darstellung von Informationen verwendet wird. Sie ist ideal geeignet, um große Mengen hierarchisch strukturierter Daten übersichtlich und platzsparend auf einer Fläche zu präsentieren. Wie der Name 'Treemap' vermuten lässt, leitet sich diese Visualisierungsmethode aus der Graphentheorie ab. Denn auch hier werden sogenannte 'gewurzelte Bäume' zur bildlichen Darstellung von Datenstrukturen verwendet. Ein gewurzelter Baum besteht aus Knoten und Kanten und stellt einen gerichteten Graphen dar. Dabei ist jeder Knoten von der Wurzel durch genau einen Pfad erreichbar. Die Knoten stellen die einzelnen Datenfelder aus einem Datensatz dar. Die Kanten sind gerichtet und spiegeln so die Eltern-Kind-Beziehungen zwischen den Knoten und Teilbäumen wider. Bei der Modellierung eines gewurzelter Baums werden ausgehend von der Wurzel mehrere gleichartige Objekte miteinander verkettet (Abbildung 2) [6]. Dadurch entstehen mehrere Ebenen, mit deren Hilfe sich hierarchische Beziehungen erzeugen lassen. Ähnlich dazu lassen sich mit einer Treemap Informationen hierarchisch auf mehreren Ebenen darstellen. In diesem Kapitel werden die Grundlagen im Bereich der Visualisierung mit Treemaps behandelt.

2.1 Treemap

Eine Treemap ist eine visuell räumliche Repräsentation eines gewurzelter Baums (engl. out-tree) [6]. Diese überträgt die hierarchischen Daten eines Datensatzes in Form von Flächen auf eine Karte (engl. map).

2 Grundlagen

Ein Baum besitzt eine ähnliche hierarchische Struktur wie eine Treemap. Ein Treemap-Element wird jedoch abweichend von einem Baumknoten mit einer gewichteten Fläche dargestellt. Jede Fläche repräsentiert dabei entweder die Wurzel, einen Teilbaum oder einen Blattknoten eines Baumes. Die Wurzel einer Baumstruktur repräsentiert die Grundfläche in einer Treemap, in der die Flächen der Kindknoten angeordnet werden. Die Flächengrößen aller Knoten werden durch die Angaben von Gewichtungen in einem Datensatz festgelegt. Unterschiedliche Elemente eines Datensatzes können daher verschieden große Flächen in einer Treemap belegen. Darüber hinaus können die Elemente mit Identifikatoren aus einem Datensatz gekennzeichnet werden oder selbst Daten innerhalb ihrer festgelegten Fläche anzeigen (Abbildung 1).

Im weiteren Verlauf dieser Bachelorarbeit werden für die Elemente einer Treemap, zum besseren Verständnis, weitgehend die Begriffe aus der Graphentheorie verwendet. Das heißt, eine Fläche wird entweder als ein Baumknoten oder als ein Treemap-Element referenziert.

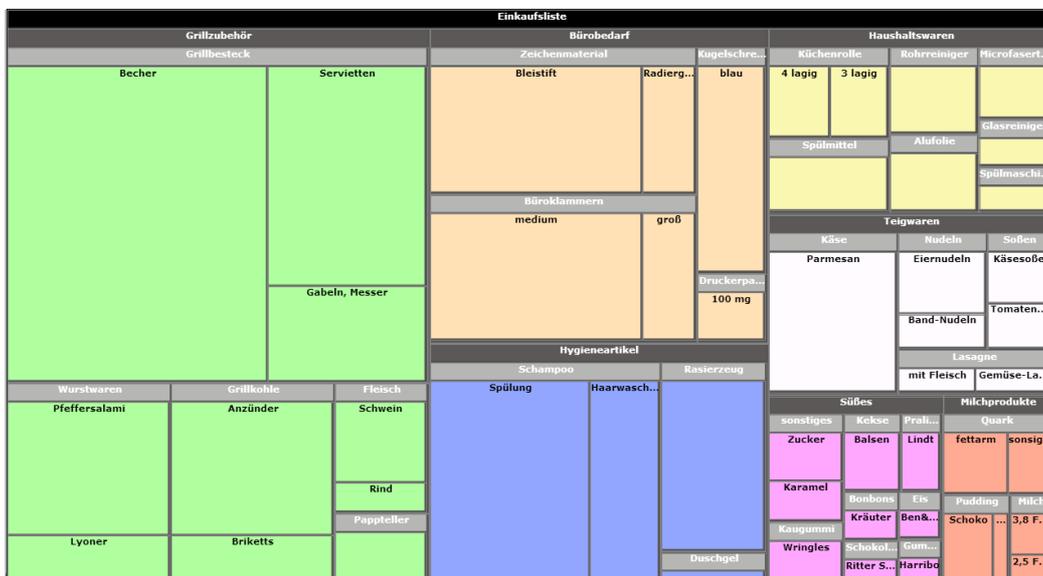


Abbildung 1: Treemap

Abbildung 2 zeigt das Resultat der Modellierung einer Treemap und eines Baumes aus einem hierarchischen Datensatz. Dabei sind Ähnlichkeiten im Aufbau beider Visualisierungen zu erkennen. Jeder Knoten in einem Baum oder jedes Element eines Datensatzes repräsentiert jeweils ein Rechteck in einer Treemap (Abbildung 2) [7]. Die Hierarchien werden in einem Baum durch gerichtete Kanten und in einer Treemap durch geschachtelte Rechtecke dargestellt. Der Aufbau einer Treemap findet durch einen rekursiven Layout-Algorithmus statt. Dieser unterteilt die Grundfläche und die Flächen der Vaterknoten in kleinere Flächen. Es existieren mehrere unterschiedliche Aufteilungsalgorithmen, welche jeweils zu einem anderen Treemap-Layout führen.

In dem Algorithmus von Shneiderman werden die Flächen der Elemente abwechselnd horizontal und vertikal auf einer Grundfläche verteilt [2]. Diese können auch ineinander geschachtelt werden. Dadurch entstehen unterschiedliche Ebenen (engl. levels) auf denen sich die einzelnen Elemente befinden. Eine Treemap kann somit aus einem hierarchischen Datensatz modelliert werden. Die Daten werden dabei (je nach Hierarchie) als Flächen den einzelnen Vaterknoten bzw. Ebenen in einer Treemap zugeteilt. Dieses Konzept ist vorteilhaft, da dadurch eine platzsparende und übersichtliche Visualisierung entsteht.

Ein weiterer Vorteil einer Treemap besteht darin, dass ein Mensch Bilder besser erfassen und verarbeiten kann als einen Datensatz, der meist in einer Tabelle dargestellt wird [3]. Demnach ist er in der Lage, Relationen zwischen verschiedenen Elementen in einer Treemap zu erkennen. Dies ist selbst dann möglich, wenn sich die Elemente auf unterschiedlichen Hierarchieebenen in einer Treemap befinden [3].

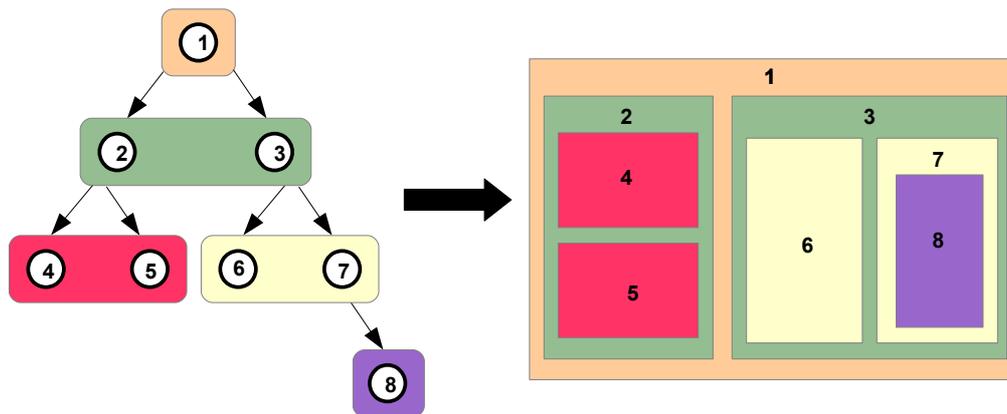


Abbildung 2: Vergleich Baum vs. Treemap [7]

2.1.1 Hierarchieebenen

Treemaps werden aus Datensätzen erstellt, die eine hierarchische Struktur besitzen. Wird ein Datensatz in Form einer Datentabelle dargestellt, dann repräsentiert jedes Element einer Treemap einen Spalteneintrag in einer Datentabelle (vgl. Abbildung 4) [8]. Die Hierarchien eines Datensatzes werden als ineinander geschachtelte Rechtecke dargestellt. Die Rechtecke befinden sich auf unterschiedlichen Ebenen. Eine Ebene wird durch die Länge des Pfades von einem Knoten bis hin zur Wurzel definiert. Die Wurzel befindet sich auf der niedrigsten Ebene (Level 0). Im Gegensatz dazu befinden sich ihre Kindknoten auf höher gelegenen Ebenen (Level > 0) (Abbildung 3). Die Höhe oder Tiefe einer Treemap wird durch den Blattknoten festgelegt, der sich auf der höchsten Ebene befindet.

Des Weiteren können Hierarchieebenen in einer Treemap, der besseren Übersichtlichkeit wegen, durch unterschiedliche Farbgebung gekennzeichnet werden (Abbildung 1, 3). Unterschieden werden hierbei Treemaps, bei denen die Elemente aller Hierarchieebenen im Verhältnis zueinander betrachtet werden und solche Treemaps, bei denen nur Elemente bis zu einer bestimmten Ebene angezeigt werden.

Letztere Visualisierung bietet bei großen Datenmengen eine bessere Übersicht. Werden nicht alle Elemente einer Treemap in einer Ebene angezeigt, dann existiert meistens eine Funktion, um den Inhalt der Knoten detaillierter anzuzeigen. Die Wurzel enthält alle Elemente und stellt die Basis der Treemap dar.

Wenn ein Element zur detaillierteren Ansicht ausgewählt wird, dann stellt dieses die neue Wurzel bzw. die neue Basis der Treemap dar. In der Treemap werden dabei nur noch die Kindknoten bzw. Nachfolger dieses Elements angezeigt, die sich auf höher gelegenen Hierarchieebenen befinden.

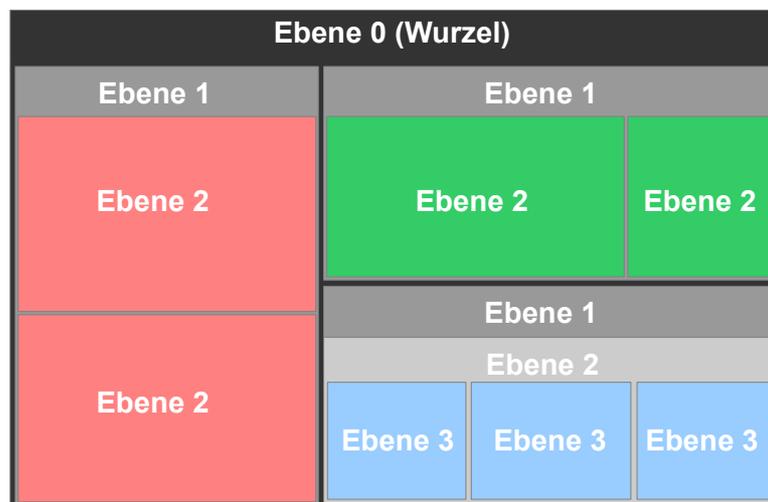


Abbildung 3: Hierarchieebenen und Farbgebung in einer Treemap

Um danach die Basis wieder in eine niedrigere Hierarchieebene zurückzuführen, müssen zusätzliche Funktionen existieren (z.B. eine 'breadcrumb navigation' oder eine entsprechende Maus-/Tastenbelegung). Eine 'breadcrumb navigation' eignet sich bei der Verwendung im Treemap-Kontext sehr gut, da sie ein einfaches Entwurfsmuster darstellt.

Sie zeigt dem Benutzer an, in welcher Hierarchieebene er sich gerade befindet und ermöglicht ihm die Navigation zu einer niedrigeren Hierarchieebene. Grundsätzlich ist es wichtig, darauf zu achten, dass eine möglichst einfache Datenexploration gewährleistet wird.

2.1.2 Flächen

Eine Treemap stellt die Daten eines hierarchischen Datensatzes als rechteckige Flächen dar. Die meisten Treemaps verfügen über eine festgelegte Grundfläche, die durch einen Layout Algorithmus in beliebig viele kleinere Flächen unterteilt werden kann. Danach können diese Flächen rekursiv weiter unterteilt werden. Dadurch werden die neu entstandenen Flächen überlagert. Die Größe jeder Fläche wird mithilfe einer quantitativen Variable festgelegt, die in dem zur Treemap zugehörigen Datensatz spezifiziert wird. Diese Variable wird auch als Gewichtung bezeichnet. Um eine Treemap erzeugen zu können, muss jedes Element eines Datensatzes eine solche Gewichtung haben, die zusätzlich bei allen Elementen gleich definiert wurde. Ist dies nicht der Fall, so kann eine Fläche entweder nicht angezeigt werden oder die einzelnen Flächengrößen beziehen sich auf Gewichtungen mit unterschiedlichen Bedeutungen. In diesem Fall können die Elemente nicht nach ihrer Flächengröße miteinander verglichen werden, sodass die Treemap an Aussagekraft verliert.

Bei der Erstellung einer Treemap müssen folgende Richtlinien bzgl. der Flächen beachtet werden. Diese Richtlinien wurden bei der Erstellung des ersten Treemap-Algorithmus von Shneiderman festgelegt [2]:

- Jedem Element (Knoten, Rechteck) eines Datensatzes wird eine Fläche zugeordnet. Die Größe dieser Fläche hängt von der Gewichtung ab.

- Ein Knoten besitzt beliebig viele Kinder, maximal jedoch nur einen Vaterknoten. Ein Blattknoten enthält keine weiteren Knoten.
- Die Gewichtung eines Knoten entspricht der Summe der Gewichtung aller Kind- oder Nachfolgeknoten. Die Flächengrößen der Treemap-Elemente sind über die Hierarchieebenen additiv [9].
- Ist ein Element A eines Datensatzes, Vorgänger eines Elements B, dann überlagern sich die zugehörigen Flächen der Rechtecke in einer Treemap. Zudem schließt das Rechteck von Element A das Rechteck von Element B komplett mit ein oder entspricht genau dem Rechteck von B [3].

Eine Treemap bietet zudem die Möglichkeit, große Datenmengen auf einer beschränkten Fläche darzustellen und direkt zu vergleichen. Dies ist vorteilhaft, da bei einer Visualisierung mit einer Baumstruktur im schlechtesten Fall (bei großen Datenmengen) eine große Darstellungsfläche benötigt wird. Bei einer Treemap hingegen wird die gesamte Fläche effizient genutzt, ohne dass sich diese bei der Expansion durch neue Elemente in ihrer Größe verändert.

Die Flächen in einer Treemap können zudem mit unterschiedlichen Farben belegt werden. Diese werden entweder im Datensatz oder im Quellcode einer Anwendung spezifiziert. Für die Wurzel und die Vaterknoten werden bei der Farbwahl meist Grautöne verwendet. Damit wird signalisiert, dass diese weitere Kindknoten enthalten. Im Gegensatz dazu können die Blattknoten beliebig eingefärbt werden. Grundsätzlich werden jedoch Blattknoten, die dem selben Vaterknoten angehören, mit der gleichen Farbe belegt. Außerdem ordnen viele Treemap-Algorithmen die Flächen sortiert nach ihrer Größe an (Abbildung 1). Die Farbgebung und die Sortierung dienen vor allem der Übersicht und dem schnellen Auffinden von Elementen in einer Treemap.

2.1.3 Daten

Herkömmliche Datensätze werden oft in einem relationalen Datenbankmodell bzw. in einem Tabellenformat dargestellt. Ein Datensatz besteht dabei aus Informationen, die z.B. von verschiedenen Instanzen oder Abteilungen eines Unternehmens generiert worden sind. Einfache Datentabellen eignen sich nicht zur Wiedergabe hierarchischer Strukturen, die in einem Datensatz auftreten können. Treemaps hingegen werden aus hierarchischen Datensätzen modelliert und können so Hierarchien anschaulich wiedergeben. Eine Treemap kann dabei entweder aus einer Liste (z.B. im JSON-Format) oder einer speziell angepassten Tabelle modelliert werden. In beiden Fällen müssen die Daten jedoch hierarchisch angeordnet sein. Ein Datensatz, der diese Anforderungen nicht erfüllt, muss erst in ein hierarchisches Datenformat umgewandelt werden.

Bei einer Tabelle werden die Daten in Zeilen und Spalten dargestellt. Eine Zeile kann auch als Tupel bezeichnet werden und beschreibt jeweils ein Treemap-Element mit seinen Kindern oder Nachfolgern. Die Spalten hingegen repräsentieren die Kategorien eines Datensatzes. Diese sind mit den unterschiedlichen Hierarchieebenen einer Treemap vergleichbar. Zudem können Spalten Informationen über das Layout enthalten, die zur korrekten Darstellung eines Elementes benötigt werden.

Eine hierarchische Liste teilt allen Daten eine eindeutige ID zu, sofern diese nicht bereits über eine solche ID verfügen. Diese ID stellt sicher, dass den Datenelementen einer Treemap die richtigen Eigenschaften aus einem Datensatz zugewiesen werden. Jedes Element in einer solchen Liste enthält zudem direkte Informationen zu seinen Kindern/Nachfolgern. Durch diese Eigenschaft wird die hierarchische Struktur bzw. die Eltern-Kind-Beziehungen der Liste und der späteren Treemap aufgebaut.

Zusätzlich zur ID jedes Elements enthält ein Datensatz in diesem Listenformat auch Layoutinformationen, wie z.B. das Farbschema oder die Gewichtung. Zudem können weiter Daten integriert werden, die nicht für das Layout einer Treemap bestimmt sind. Diese Daten dienen dem Benutzer nur zu Informationszwecken und können in der späteren Implementierung in der Treemap angezeigt werden (z.B. innerhalb der Fläche oder bei der Auswahl eines Elements) (Abbildung 4). Minimal müssen jedoch für jedes Element in einem Datensatz zumindest eine eindeutige ID und die Gewichtung zur Bestimmung der Flächengröße vorhanden sein.

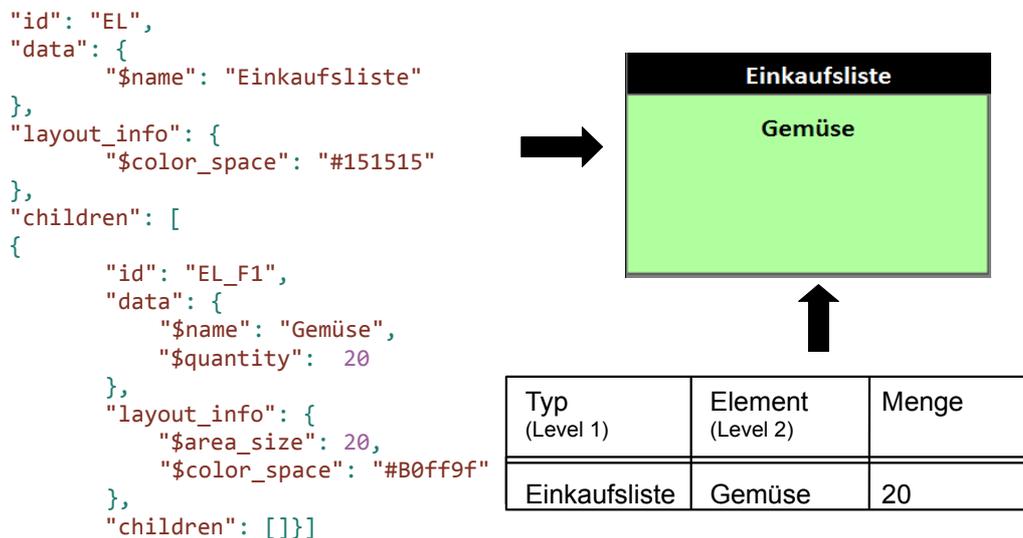


Abbildung 4: Hierarchische Liste (JSON) und Treemap-Element

2.1.4 Treemap-Algorithmen

Eine Treemap wird durch einen rekursiven Algorithmus erstellt. Dieser ist für die Aufteilung der Flächen in weitere untergeordnete Flächen zuständig. Je nach Layout Algorithmus entstehen dabei Treemaps mit unterschiedlichen Flächenformen.

Somit existieren z.B. runde, rechteckige oder dreieckige Flächen [2]. Die Aufteilung durch einen Layout Algorithmus sollte so durchgeführt werden, dass keine 'Löcher' in der Treemap entstehen. 'Löcher' bezeichnen dabei Flächen, die keinem Element in einem Datensatz zugeordnet sind.

Seit der Erfindung der Treemap wurden viele neue und unterschiedliche Layout-Algorithmen entwickelt. In dieser Bachelorarbeit werden jedoch nur Algorithmen betrachtet, die Treemaps mit rechteckigen Flächen erstellen.

Ein Treemap-Algorithmus wird unter anderem anhand folgender Vergleichskriterien bewertet: den Seitenverhältnissen (engl. aspect ratio), der Stabilität und der Anordnung der Elemente im Layout.

Die Division der Höhe durch die Breite einer Fläche bezeichnet das Seitenverhältnis eines Elements. Je besser die Seitenverhältnisse in einer Treemap sind, desto stärker nähern sich deren Flächen einer quadratischen Form an. Bei schlechten Seitenverhältnissen entstehen langgezogene Flächen, die den Größenvergleich der Elemente in einer Treemap erschweren.

Die Stabilität beschreibt wie stark ein Element dazu tendiert seine Position innerhalb der Treemap zu verändern. Dieses Vergleichskriterium ist vor allem bei interaktiven Treemaps wichtig, bei denen sich die Flächengrößen indirekt oder direkt in der Visualisierung verändern lassen [10]. Je 'stabiler' eine Treemap ist, desto weniger ändert sich die Position eines Elements bei der Veränderung seiner Flächengröße. Der Benutzer verliert daher das Element bei einer Änderung nicht aus den Augen und bewahrt so den Überblick über den Datensatz. Denn dieser tendiert dazu, sich die Positionen der Elemente in einer Treemap zu merken, um diese bei einem wiederholten Zugriff schnell wiederzufinden.

Die Anordnung überprüft, ob die natürliche Ordnung eines Datensatzes in einer Treemap erhalten bleibt. Mit anderen Worten, es wird überprüft, ob die Nachbarschaftsverhältnisse der Elemente einer Treemap und die Anordnung der Elemente in einem Datensatz übereinstimmen. Auch dieses Kriterium dient der Übersicht und erleichtert den direkten Vergleich der Treemap-Struktur mit der Struktur des Datensatzes.

Im weiteren Verlauf dieses Kapitels wird der Squarified-Algorithmus vorgestellt, da dieser für die Implementierung des Prototyps dieser Bachelorarbeit verwendet wurde.

Der Squarified-Algorithmus teilt, wie der Name schon andeutet, die Flächen einer Treemap in Rechtecke auf. Deren Form soll dabei möglichst quadratisch sein. Das heißt, es wird darauf geachtet, dass die Seitenverhältnisse der Flächen einer Treemap möglichst optimal sind. Das Ergebnis ist eine übersichtliche und benutzerfreundliche Visualisierung. Der Algorithmus lässt sich rekursiv auf alle Hierarchieebenen anwenden. Dieser verfährt, ausgehend vom größten Element bis hin zum kleinsten Element eines Datensatzes, wie in Abbildung 5 dargestellt.

Am Anfang ist eine Grundfläche gegeben, in der sich noch keine Elemente befinden. Die einzufügenden Elemente werden in einer Liste gespeichert. Die Fläche des Rechtecks eines Elements wird nun so eingefügt, dass ihr Seitenverhältnis möglichst ausgeglichen ist. Das Ziel ist es, eine quadratische Form der einzufügenden Fläche zu erreichen. Der Algorithmus erstellt beim Einfügen eines Elements horizontale oder vertikale Reihen. Diese werden ebenfalls in einer Liste gespeichert. Die Elemente werden nun in Form von Rechtecken in diese erzeugten Reihen eingefügt.

Eine Reihe wird dabei vertikal angelegt, wenn die Fläche des Vaterknotens, in die ein neues Element eingefügt werden soll, breiter als hoch ist (Abbildung 5, Schritt 1). Ist das Gegenteil der Fall, dann wird eine Reihe horizontal angelegt. Ein Element wird entweder in eine bereits bestehende Reihe eingefügt oder es wird eine neue Reihe in der noch freien Fläche des Vaterknotens angelegt. Wenn sich die Seitenverhältnisse der bestehenden Flächen beim Einfügen in die aktuelle Reihe verbessern, dann wird das neue Element in diese Reihe eingefügt. Ist das Gegenteil der Fall, dann wird eine neue Reihe erstellt, in welche das Element eingefügt wird. Auch hier wird dann nach obigen Kriterien überprüft, ob diese Reihe vertikal oder horizontal angelegt werden muss [9]. Unter der Verbesserung des Seitenverhältnisses eines Elements ist die Veränderung der Flächenform hin zu einer mehr 'quadrat-ähnlicheren' Form zu verstehen. Das heißt, dass das Seitenverhältnis einer Fläche stärker an das Seitenverhältnis eines Quadrats angenähert wird. Ein Quadrat besitzt dabei ein Seitenverhältnis von 1 [9].

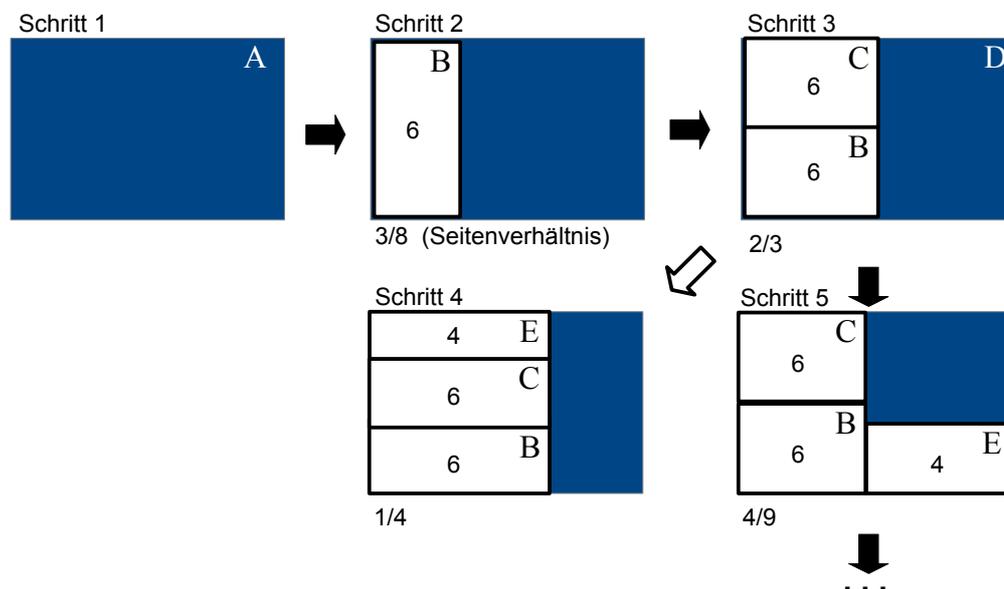


Abbildung 5: Squarified-Algorithmus [11]

Abbildung 5 und 6 beschreiben den Ablauf des Squarified-Algorithmus [9]. Zuerst wird die Fläche des Vaterknotens untersucht (Schritt 1). Da diese breiter als hoch ist, wird eine vertikale Reihe angelegt. In diese wird das erste Element eingefügt (Schritt 2). Das zweite Element wird nun ebenfalls in diese Reihe eingefügt, da sich die Seitenverhältnisse des ersten Elements dadurch verbessern (Schritt 3). Das dritte Element wird nun in die aktuelle Reihe eingefügt. Dadurch verschlechtern sich die Seitenverhältnisse der vorhandenen Elemente (Schritt 4). Deshalb wird nun eine neue Reihe in der noch freien Fläche D erzeugt. Die Fläche D ist höher als breit. Somit wird eine neue horizontale Reihe erzeugt und das dritte Element eingefügt (Schritt 5) [11].

Der oben beschriebene Algorithmus kann nun auf die restlichen Elemente der Liste entsprechend angewandt werden. Dabei können diese auch in bereits bestehende Knoten eingefügt werden. Dadurch entstehen neue Hierarchieebenen in einer Treemap.

```
squarify(Queue nodes) {
    Queue currentRow;
    nodes.sort(); // Elemente nach Größe sortieren

    while (nodes.length > 0) {
        Item current := nodes.dequeue();
        // Seitenverhältnisse verbessert – in aktuelle Reihe einfügen
        if (worst(currentRow + current) < worst(currentRow)){
            currentRow.enqueue(current);
        // Seitenverhältnisse verschlechtert – neue Reihe erstellen
        } else {
            layoutRow(currentRow);
            currentRow.clear();
            currentRow.enqueue(current);
        }
    }
}

foreach (Item node in nodes) {
    squarify(node.children);
}
```

Abbildung 6: Squarified-Algorithmus, Pseudocode [12]

Der Squarified-Algorithmus erzeugt eine Treemap, deren Elemente optimierte Seitenverhältnisse haben. Es wird daher beim Einfügen eines neuen Elements immer die Position ausgewählt, in der sich die optimalen Seitenverhältnisse ergeben (unter Berücksichtigung der bereits vorhandenen Elemente) [12]. Beim Verändern der Flächengröße eines Elements kommt es daher oft zu Positionsverschiebungen. Darum wird die Stabilität einer Squarified Treemap als mittelmäßig eingestuft (Abbildung 7) [13]. Zudem wird die natürliche Ordnung der Daten in einer Squarified Treemap nicht gleichermaßen wiedergegeben wie in dem Datensatz aus welchem diese besteht. Daher ist die Anordnung der Elemente eines Datensatzes in einer Squarified Treemap, im Vergleich zu anderen Treemap Visualisierungen schlecht (Abbildung 7) [13].

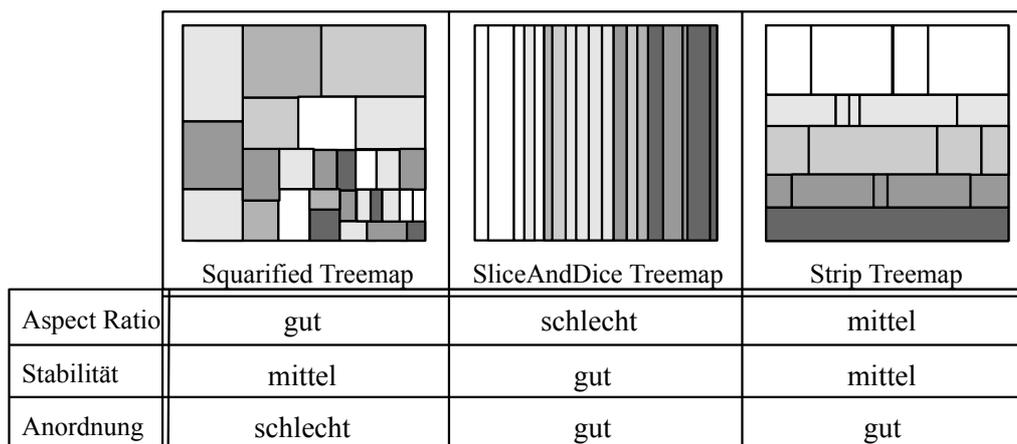


Abbildung 7: Vergleich Treemap-Algorithmen [13]

Es gibt keinen Algorithmus, der in allen Vergleichskriterien ein gutes Ergebnis liefert. Daher muss bei der Auswahl eines Algorithmus (je nach Anwendungsfall) zwischen dem Nutzen der oben aufgeführten Vergleichskriterien abgewogen werden.

Ein SliceAndDice-Algorithmus bietet z.B. eine bessere Anordnung der Elemente und eine bessere Stabilität als ein Squarified-Algorithmus, jedoch auch ein schlechteres Seitenverhältnis der Elemente [14]. Im Vergleich dazu bietet ein Strip-Algorithmus eine gute Anordnung, allerdings nur ein mittelmäßiges Seitenverhältnis sowie eine mittelmäßige Stabilität (Abbildung 7) [14].

3 Treemaps im Unternehmenskontext

Treemaps werden zur Visualisierung großer Datenmengen eingesetzt. Die Daten werden platzsparend und in einer hierarchischen Anordnung präsentiert. Eine Treemap ist einfach zu implementieren und bietet eine übersichtliche und detaillierte Darstellung. In einem Unternehmen kann diese in vielen Anwendungsbereichen eingesetzt werden, wie z.B. zur Datenpräsentation, -analyse oder auch als Entscheidungshilfe zu Planungszwecken. Zur Darstellung großer Datenmengen werden in Unternehmen vor allem bewährte Visualisierungsmethoden eingesetzt. Meist handelt es sich dabei um Kreis-, Säulen-, Balken- oder Liniendiagramme [15] [16] [17]. Diese sind jedoch oft unübersichtlich und daher ungeeignet (Abbildung 8). Zudem können diese Visualisierungen nur unzureichend widerspiegeln, welche Auswirkungen eine Wertänderung auf andere Bereiche eines Datensatzes hat. Treemaps eignen sich im Gegensatz dazu, besonders gut für Visualisierungszwecke in einem Unternehmen.

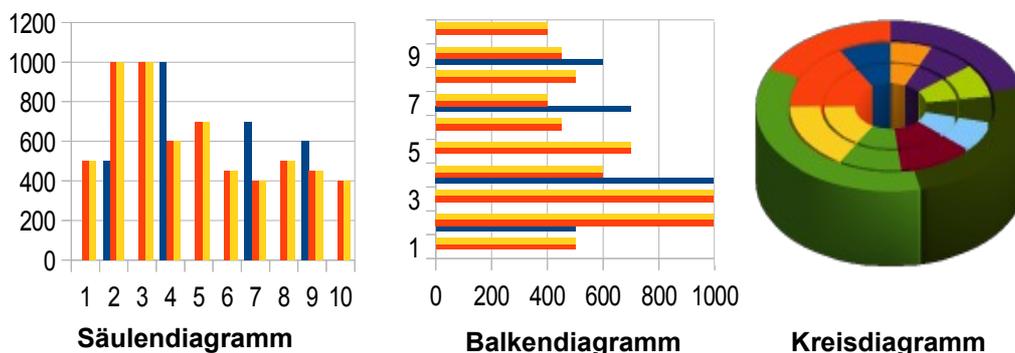


Abbildung 8: Diagrammtypen

Im Rahmen dieser Bachelorarbeit wurde aufgrund dessen ein Konzept entworfen, das auf einer Treemap basiert und die Strategieplanung in einem Unternehmen unterstützt. Dieses Konzept soll zur internen Kosten- und Budgetplanung eingesetzt werden. In den folgenden Abschnitten werden das Einsatzgebiet, die Funktionsanforderungen und die Designkriterien einer solchen Visualisierung beschrieben. Zudem wird erörtert, welche Algorithmen sich am besten für die Erstellung von Treemaps in diesem Bereich eignen.

3.1 Einsatzgebiet

Das Endprodukt des Treemap-Prototyps dieser Bachelorarbeit soll später in einem Unternehmen eingesetzt werden. Ein Unternehmen besteht meistens aus mehreren unterschiedlichen Abteilungen. Jede Abteilung hat dabei einen oder mehrere Aufgabenbereiche. Durch diese Aufgaben entstehen Kosten in den Abteilungen. Jede Abteilung bekommt dadurch vom Management ein bestimmtes Budget zugeteilt, mit dem die entstehenden Kosten gedeckt werden können. In einer Abteilung können unterschiedliche Kosten entstehen. Zum einen existieren Kosten, die durch Dienstleistungen des Personals entstehen (z.B. im Rahmen eines Projektes). Die Personalkosten haben meistens den größten Anteil am Budget einer Abteilung und bestehen aus den Kosten für Gehälter, Löhne, Lohnnebenkosten und soziale Anwendungen [18]. Zum anderen entstehen Sachkosten, die die Kosten für materielle und immaterielle Mittel und Leistungen, die für die Bereitstellung von Dienstleistungen oder Gütern benötigt werden, beschreiben [18].

Die Kosten aller Abteilungen werden in einem oder mehreren Datensätzen gesammelt (vgl. Abbildung 10). Aus diesen Daten werden Visualisierungen für das Management eines Unternehmens erstellt.

Dadurch kann das Gesamtbudget eines Unternehmens im Zuge der Strategieplanung auf die einzelnen Abteilungen verteilt werden. Dabei ist zu beachten, dass eine Abteilung möglichst genug finanzielle Mittel erhält, um ihre Kosten zu decken. Die Budgetgröße ist demnach meistens von den entstehenden Kosten in einer Abteilung abhängig.

Am Anfang der Kostenplanung wird eine Kostenanalyse durchgeführt. Diese bezieht sich auf einen festgelegten Zeitraum oder ein Quartal. Die aktuelle Kostensituation einer Abteilung wird anhand des Kostenniveaus (absolute Höhe der Kosten), der Kostenstruktur (Anteil der fixen und variablen Kosten) und des Kostenverlaufs (Reaktion auf Kostentreiber) erfasst [14]. Anhand dieser Analyse lässt sich dann der Umfang der zukünftigen Kosten abschätzen. Die Kosten einer Abteilung entstehen zum größten Teil durch die oben beschriebenen Personal- und Sachkosten. Das Ergebnis der Kostenanalyse ist ein Kostenplan, der als Strategieplan bezeichnet werden kann [18]. Dieser wird meist für einen festgelegten Zeitraum oder ein Quartal erstellt. Die geschätzten Kosten können sich in einem Planungszeitraum durch unvorhersehbare Ereignisse verändern (z.B. durch den Ausfall von Mitarbeitern oder durch das Überschreiten der Abschlussfrist eines Projekts).

Die Strategieplanung ist ein mehrstufiger Prozess, der nicht nur einmal, sondern mehrfach durchgeführt werden muss [18]. Ziel der Strategieplanung ist es, durch regelmäßige Kontrolle der Kosten, Abweichungen von den Vorgaben im Kostenplan möglichst frühzeitig vorzusehen und den Strategieplan dementsprechend anzupassen [14]. Deshalb wird überprüft, ob mit der aktuellen Kostenentwicklung die Zielvorgaben im Strategieplan noch erreicht werden können. Dadurch können Bereiche, in denen Handlungsbedarf zur Kostenreduktion herrscht, schnell identifiziert werden. Eine Handlung beschreibt entweder die frühzeitige Anpassung des Budgets im Strategieplan oder das Eingreifen, z.B. in die Durchführung oder Planung eines Projektes.

Letzteres ist der Fall, wenn z.B. ein Anstieg der Projektkosten droht, der eine festgelegte Grenze im Strategieplan überschreitet.

3.2 Funktions- und Designanforderungen

Treemaps, die in einem Unternehmen zur Darstellung von internen Geschäftsinformationen eingesetzt werden, müssen bestimmte Voraussetzungen sowohl im Funktionsumfang als auch im Design erfüllen. Im Folgenden werden einige dieser Funktionen und Designkriterien aufgeführt. Diese sind jedoch stark vom jeweiligen Anwendungsgebiet abhängig. Eine Treemap, die in einem Unternehmen eingesetzt wird, kann als 'Business-Treemap' [19] bezeichnet werden. Im folgenden Kapitel werden die Anforderungen behandelt, die eine Business-Treemap erfüllen muss. In diesem Zusammenhang werden nur die Aspekte beschrieben, die im Bereich der Kosten-/Budgetplanung, Umsatzdarstellung oder Personalverwaltung benötigt werden.

Eine Treemap-Anwendung muss grundsätzlich auf dem Softwaresystem und der vorhandenen Hardware eines Unternehmens ausführbar sein. Folglich muss ein Austausch von Daten zwischen dem System und der Anwendung gewährleistet werden [20].

Obendrein wird eine Business-Treemap aus Daten modelliert, die sich stets verändern können. Daher müssen sich die Treemap-Elemente möglichst zeitnah den Wertänderungen eines Datensatzes anpassen. Zudem ist es hilfreich, wenn ein Benutzer die Daten direkt in der Treemap verändern kann.

Ferner werden für die Wiederverwendbarkeit eines geänderten Datensatzes, Funktionen zum Speichern und Laden der Daten benötigt.

Eine Business-Treemap sollte zudem eine möglichst gute Datenexploration ermöglichen. Dadurch muss eine detaillierte Darstellung der Daten aller Hierarchieebenen gewährleistet werden, denn bei großen Datensätzen können die Elemente der höheren Ebenen ausgeblendet werden. Des Weiteren werden Elemente mit kleinen Gewichtungen in der Treemap sehr klein und daher unkenntlich dargestellt.

Innerhalb der Fläche eines Treemap-Elements können auch Informationen dargestellt werden. Dabei besteht die Möglichkeit, irrelevante Informationen erst bei der Auswahl eines Elements in einem externen Fenster oder einem Tooltip anzuzeigen. Bei großen Datensätzen wird so verhindert, dass die Treemap unübersichtlich wird.

Die Designanforderungen einer Business-Treemap orientieren sich weitgehend an den Vergleichskriterien aus dem vorherigen Kapitel. Grundsätzlich müssen alle Treemap-Elemente gute Seitenverhältnisse aufweisen, damit die Flächengrößen einzelner Elemente ohne Aufwand miteinander verglichen werden können. Quadratische Flächen in einer Treemap führen zu einer besseren Übersicht und somit zu einer besseren Benutzerfreundlichkeit (engl. usability).

Eine Treemap ist bei der Strategieplanung in einem Unternehmen nicht der einzige Entscheidungsträger. Daher werden oft andere Visualisierungen (Abbildung 8) und der Datensatz (Abbildung 10) zu Analysezwecken verwendet. Dementsprechend kann es vorkommen, dass einzelne Visualisierungsformen miteinander oder mit dem Datensatz verglichen werden. Deshalb sollte die Anordnung der Elemente in einer Treemap und in allen anderen Visualisierungen, weitgehend mit der Anordnung der Daten im Datensatz übereinstimmen.

Business-Treemaps müssen eine gute Stabilität aufweisen, da diese aus Daten modelliert werden, die sich häufig ändern [21].

Ein Benutzer merkt sich oft die Position der Elemente und bildet so ein mentales Bild von der Treemap. Verändert ein Element nun seine Position, dann besteht die Gefahr, dass dieser den Überblick verliert. Dies kann vor allem bei großen Datensätzen der Fall sein.

Die Flächen einer Treemap können mit beliebigen Farben belegt werden. Farbmuster können dabei auf bestimmte Eigenschaften der Elemente verweisen oder zur Differenzierung der Knoten verwendet werden. Zudem können Änderungen im Datensatz visuell in der Treemap dargestellt werden. Die Farben sollten so gewählt werden, dass keine Missverständnisse auftreten. Eine rote Fläche könnte z.B. als Verlust in einem Unternehmensbereich oder als Warnsignal und somit als Handlungsbedarf interpretiert werden. Die Farbe 'Grün' hingegen wird meist mit einer Gewinnsteigerung assoziiert (Abbildung 9) [21].

In Anbetracht dieser Funktions- und Designanforderungen und der im Kapitel 2.1.4 vorgestellten Algorithmen eignet sich ein modifizierter Strip oder Squarified-Algorithmus für die Darstellung unternehmensrelevanter Informationen am besten. Ein Strip-Algorithmus kann dabei so angepasst werden, dass dieser eine Treemap mit guten Seitenverhältnissen und einer guten Stabilität erzeugt. Im Gegensatz dazu, muss ein Squarified-Algorithmus so verändert werden, dass dieser eine Treemap mit einer guten Stabilität und Anordnung erstellt. Ein SliceAndDice-Algorithmus stellt die Elemente einer Treemap als langgezogene Streifen dar. Dieser Ansatz eignet sich deshalb im Unternehmensbereich eher weniger, da eine übersichtliche Darstellung nicht gewährleistet wird (vgl. Abbildung 7).

Die Abbildung 9 zeigt ein Beispiel einer Business-Treemap (vgl. Abbildung 17). Diese stellt den Absatz eines Quartals an den verschiedenen Verkaufsstandorten eines Unternehmens dar. Die dunkelgrauen Flächen beschreiben die Unternehmensstandorte.

Die farbigen Flächen kennzeichnen hingegen die Erfüllung der erwarteten Verkaufsquote in diesem Quartal [21]. Standorte, die rot markiert sind, deuten auf eine zu niedrige Verkaufsquote hin. Grüne Flächen stellen den gegensätzlichen Sachverhalt dar. Mithilfe der Farbintensität kann zusätzlich die Höhe der Verkaufsquoten dargestellt werden.

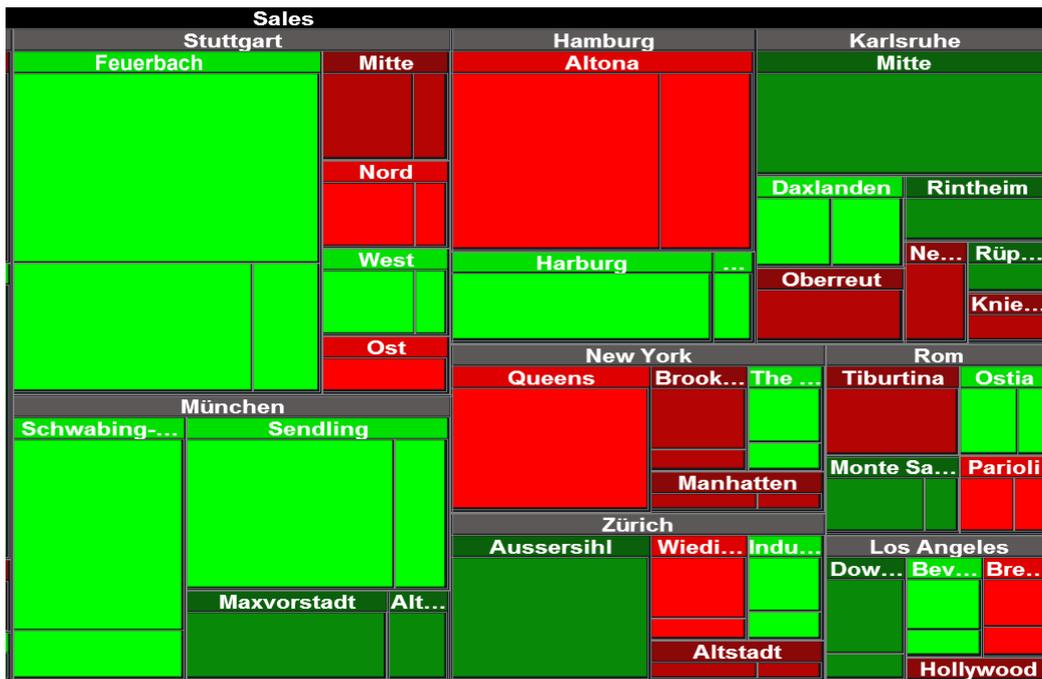


Abbildung 9: Business-Treemap

4 Implementierung

Eine Aufgabe dieser Bachelorarbeit war die Implementierung eines Treemap-Prototyps. Anhand diesem soll das Konzept veranschaulicht werden, das speziell zur Kosten-/Budgetplanung entworfen wurde.

Der erste Teil dieses Kapitels beinhaltet die Beschreibung des Konzepts. Zuerst wird das 'Grundgerüst' beschrieben, in das das Konzept eingebettet wurde. Im zweiten Teil werden die Programmiersprachen und das Toolkit beschrieben, die bei der Entwicklung des Prototyps zum Einsatz kamen. Abschließend wird der Treemap-Prototyp vorgestellt.

4.1 Konzept

Die Kosten, die in den Aufgabenbereichen der Abteilungen eines Unternehmens entstehen, werden in einem Datensatz gespeichert (Abbildung 10). Anhand diesem kann nun eine Treemap erstellt werden. Dazu wird ein Verfahren verwendet, das auf einem Squarified-Algorithmus basiert. Der Aufbau der Treemap erfolgt dementsprechend wie beim Squarified-Algorithmus. Dadurch werden rechteckige Flächen erzeugt, die optimale Seitenverhältnisse aufweisen. Wenn nun der Wert eines Elements in einer Squarified Treemap verändert wird, dann ändert sich auch dessen Position. Denn jedes Mal, wenn der Algorithmus ausgeführt wird, ordnet dieser die Elemente neu nach deren Größe an. Um dies zu verhindern und so eine möglichst stabile Treemap zu erhalten, wird bei diesem Konzept auf die Sortierung der Elemente verzichtet.

Somit kann sich der Wert eines Elements verändern, ohne dass dieses seine Position in der Treemap wechselt. Die Anordnung der restlichen Elemente bleibt damit erhalten und ist somit immer äquivalent zur Anordnung im Datensatz.

Competency	Component	Attributes			
		Sachkosten	Personalkosten	Gesamt budget	Personal
Strategie, Planung & Steuerung	IT Strategie	0	50000	50000	500
	IT Governance	5000	100000	100000	900
	IT Bebauung	0	100000	100000	900
	Innovations Manag.	0	60000	60000	600
	Business Continuity	0	70000	70000	700
	Projekt Management	0	45000	45000	400
	Personal Management	0	40000	40000	400
	Budget Management	3500	50000	50000	500
	Einkauf	2000	10000	40000	700
	IT Controlling	0	0	0	80
Anforderungs Management &	Überwachung	0	0	0	70
	Anforderungs Manag.	0	45000	45000	40
	Servicemanagement	0	60000	600000	57
	Kunden-Analysen	0	0	0	90

Abbildung 10: Beispiel eines Datensatzes im Unternehmensbereich

Mit diesem Algorithmus wird nun eine Treemap für die Strategieplanung in einem Unternehmen wie folgt aufgebaut:

Die Budgets, die die Kosten eines Unternehmens, einer Abteilung oder eines Aufgabenbereiches decken, werden als rechteckige Flächen dargestellt. Die Flächengewichtungen bzw. Flächengrößen werden dabei durch die entstehenden Kosten in den jeweiligen Bereichen eines Unternehmens festgelegt (Abbildung 10). Das Gesamtbudget eines Unternehmens besteht aus den Budgets der Abteilungen. Die Abteilungen wiederum verteilen dieses auf die einzelnen internen Aufgabenbereiche.

Dort wird das Budget zur Kostendeckung verwendet (z.B. von Personal- und Sachkosten). Die Flächen der Budgets werden, je nach Zugehörigkeit, überlagert dargestellt. Dadurch entsteht die hierarchische Struktur der Treemap. In dem Datensatz, aus dem der Treemap-Prototyp modelliert wurde, werden die einzelnen Hierarchieebenen durch Spaltenkategorien beschrieben. In Abbildung 10 wird ein Datensatz gezeigt, der die Hierarchieebenen 'Competency', 'Component' und 'Attributes' besitzt. Die Zeileneinträge hingegen beschreiben die Gewichtung der einzelnen Flächen einer Treemap. Die Umwandlung dieses Datensatzes in eine Treemap, erfolgt wie in Kapitel 2.1.3 beschrieben.

Um nun das Budget einer Abteilung in Folge der Strategieplanung zu verändern, müssen die Flächengrößen einer Treemap veränderbar sein. Das entwickelte Konzept sorgt dafür, dass alle Elemente möglichst wenige Positionsänderungen durchführen.

Bei einer Budgeterhöhung wird die Wertänderung eines Knotens jeweils an die Knoten auf den niedrigeren Hierarchieebenen weitergegeben. Dies betrifft jedoch nur die Knoten, die sich auf dem Pfad vom geänderten Knoten bis hin zur Wurzel befinden. Das heißt eine Wertänderung wird zu den Werten der Vorgänger des geänderten Elements im vollen Umfang hinzuaddiert oder subtrahiert. Außerdem wird diese gleichmäßig zwischen den direkten Nachfolgern aufgeteilt. Besitzen diese Elemente ebenfalls Nachfolger, dann wird dieser Prozess ein weiteres Mal ausgeführt. In diesem Fall wird jedoch nicht die Wertänderung, sondern das zugewiesene Teilbudget jedes Elements weiter aufgeteilt.

In Abbildung 11 wird dieser Vorgang dargestellt. Der Aufgabenbereich 'Applikationsbetrieb' in der Abteilung 'Applikationswartung' erhält eine beispielhafte Erhöhung seines Budgets um 333 €. Dadurch wird diese Wertänderung auf die Budgets der nächsthöheren Hierarchieebene gleichverteilt.

Die Budgets der Sach-, Personal- und Verwaltungskosten werden demnach jeweils um 111 € erhöht. Angenommen das Budget der Personalkosten wäre in weitere Bereiche unterteilt, dann würde die zuvor zugeteilte Budgeterhöhung von 111 €, nun auf diese Bereiche gleichverteilt werden. Des Weiteren wird das Budget der Abteilung des veränderten Aufgabenbereichs um dessen Budgetänderung von 333€ erhöht.

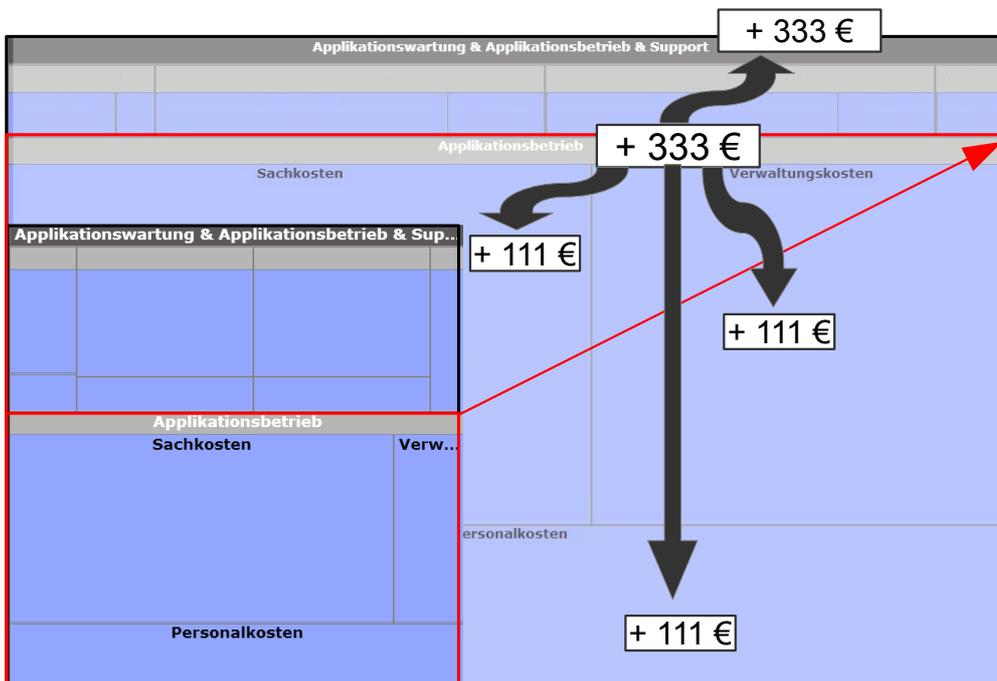


Abbildung 11: Budget-Erhöhung in einer Treemap

Wenn nun das Budget eines Elements in Folge einer Budgetänderung eines Vorgängerelements nicht verändert werden soll, dann muss dieses gesperrt werden. Dieses Verhalten kann anhand Abbildung 12 verdeutlicht werden. Hier ist das Budget der 'Sachkosten' des Aufgabenbereichs 'Applikationsbetrieb' gesperrt. Deshalb wird dieses bei einer Budgetänderung des zugehörigen Aufgabenbereichs oder der Abteilung nicht verändert.

Zusätzlich soll auch die Möglichkeit bestehen, dass das Budget eines Vorgängerelements gesperrt werden kann, sodass dieses bei der Budgetänderung eines Nachfolgeelements nicht verändert wird. In diesem Fall wird die Höhe der Wertänderung auf die Nachbarelemente verteilt, die über keine Sperrung verfügen. Ist ein Nachbarelement gesperrt, dann verändert dieses seinen Budgetwert nicht. In Abbildung 12 wird das gesperrte Element 'Applikationswartung & Applikationsbetrieb & Support' einer Treemap dargestellt. Dabei wurde das Budget eines Nachfolgeelements um 777 € verringert. Da nun das Gesamtbudget des Vorgängers, durch die Sperrung seinen Wert nicht verändern darf, wird die Wertänderung auf die nicht gesperrten Nachbarelemente gleichverteilt. Das Budget jedes Nachbarelements, das nicht gesperrt wurde und sich in derselben Hierarchieebene wie das geänderte Element befindet, wird um 111 € erhöht. Bei einer Budgetkürzung werden die neu zugeteilten Teilbudgets von den Budgets der Nachbarelemente subtrahiert.

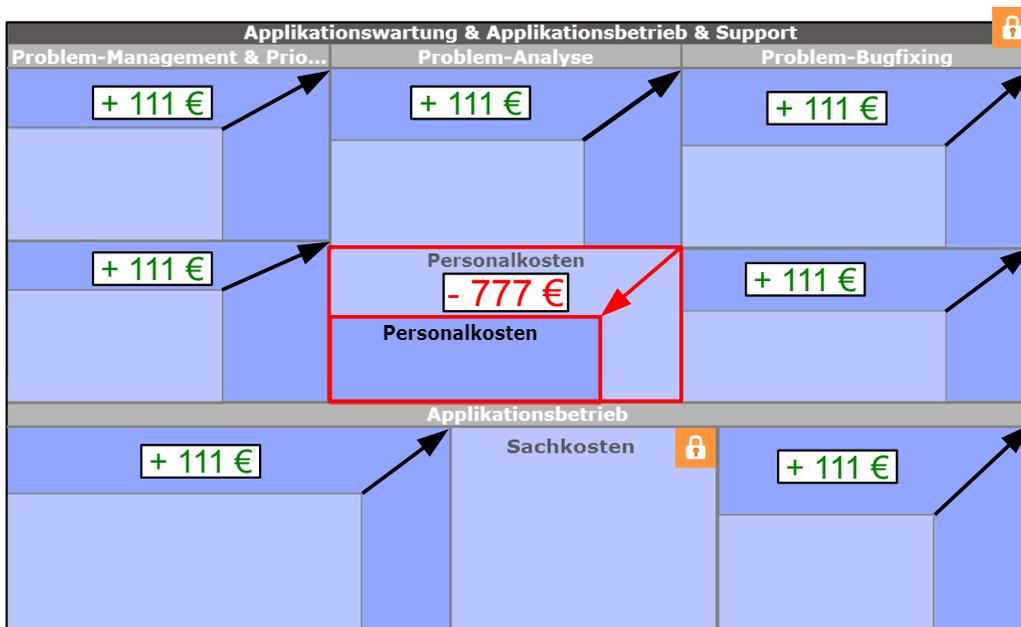


Abbildung 12: Budgetkürzung mit Budget-Sperren

Die Gewichtung bzw. das Budget eines Elements kann auch direkt in der Treemap verändert werden. Dazu stehen dem Benutzer unterschiedliche Interaktionsmöglichkeiten zur Verfügung (siehe Kapitel 4.2).

Um einen Benutzer bei der Strategieplanung mithilfe der Treemap bestmöglich zu unterstützen, ist es hilfreich, die in Kapitel 3.2 beschriebenen Funktions- und Designanforderungen zu beachten.

4.2 Umsetzung

Es gibt viele unterschiedliche Möglichkeiten, eine digitale Treemap zu erstellen. Als Erstes muss der Entwickler die Programmiersprache wählen, in der er eine Treemap erstellen möchte. Treemap-Visualisierungen lassen sich sowohl in objektorientierten Programmiersprachen (z.B. Java, C++ oder C#) entwickeln, als auch in webbasierten Programmiersprachen (z.B. Javascript, HTML und CSS). Derzeit existiert für fast jede gängige Programmiersprache ein Toolkit, mit dem sich die Entwicklung einer Treemap durch vorgefertigte Bibliotheken, Klassen und Schnittstellen vereinfachen lässt.

Der Treemap-Prototyp in dieser Bachelorarbeit wurde mithilfe eines JavaScript Toolkits namens 'InfoVis' entwickelt [22]. Das Toolkit enthält eine Bibliothek mit verschiedenen Treemap-Algorithmen, die zur Implementierung einer Squarified-, SliceAndDice- oder Strip-Treemap verwendet werden können. Die graphische Benutzeroberfläche der Treemap wurde mithilfe dieses Toolkits und den Programmiersprachen HTML5 und CSS3 erstellt. Zusätzlich wurden alle Funktionen der Treemap in JavaScript implementiert. Die Verwendung von webbasierten Programmiersprachen macht diese Anwendung plattformunabhängig. Der Prototyp kann daher über das Internet und mithilfe eines Browsers auf allen gängigen Betriebssystemen ausgeführt werden.

In diesem Kapitel wird die Umsetzung des in Kapitel 4.1 beschriebenen Konzepts in einen funktionsfähigen Prototyp beschrieben.

Der Hauptbestandteil des Prototyps ist eine Treemap, die aus einem hierarchischen Datensatz modelliert wurde. Der Datensatz enthält einschließlich der Wurzel vier Spalten. Abbildung 10 zeigt einen Ausschnitt dieses Datensatzes. Die Wurzel der Treemap beschreibt das Gesamtbudget des Unternehmens und somit die Hierarchieebene null. Die Attribute 'Personal-' und 'Sachkosten' repräsentieren jeweils die Blattknoten der Treemap und somit die Hierarchieebene drei. Beim Aufbau der Treemap überlagern die Flächen der Budgets höherer Hierarchieebenen die der niedrigeren. Da das verwendete Toolkit für den Aufbau einer Treemap eine hierarchische Liste voraussetzt, wurde der Datensatz zuerst in das Datenformat JSON (JavaScript Object Notation) umgewandelt.

Abbildung 13 zeigt die Startseite des Prototyps. Die Treemap steht dabei im Mittelpunkt der graphischen Benutzeroberfläche. In der linken Toolbar befinden sich eine 'breadcrumb navigation', die Schaltflächen zum Umschalten der Benutzeroberflächen und eine Chronik, die die vorgenommenen Änderungen protokolliert. Zusätzlich wird die aktuell vorgenommene Änderung an einem Treemap-Element in der unteren Toolbar angezeigt. Links daneben befinden sich folgende Funktionen:



Datensatz: Zeigt die Sicht auf die hierarchische Liste, aus der die Treemap modelliert wurde. Als Datenformat wird JSON verwendet.



Farbwechsel: Die Farben der Blattknoten der Treemap werden zufällig ausgewählt. Diese Funktion ermöglicht eine neue Farbverteilung.



Ausblenden: Blendet die Knoten auf den höheren Ebenen aus, sodass eine bessere Übersicht gewährleistet wird.



Speichern: Ermöglicht das Speichern eines geänderten Datensatzes. Als Speicherformat wird JSON verwendet.

Der Prototyp enthält insgesamt drei Benutzeroberflächen. Diese bieten dem Benutzer jeweils einen anderen Funktionsumfang. Die Funktionen wurden zum größten Teil direkt in die Treemap integriert. Der Benutzer kann je nach Funktionsbedarf die Oberflächen wechseln. In den folgenden Abschnitten werden das Design und der Funktionsumfang dieser Oberflächen vorgestellt.

4.2.1 Datenexploration

Die Datenexplorationsoberfläche repräsentiert die Startseite des Prototyps (Abbildung 13). Diese wurde für die Erkundung der Daten in der Treemap entwickelt. Ein Benutzer kann die Informationen auf allen Hierarchieebenen betrachten. Die Wurzel bildet die Basis dieser Betrachtung. Sie wird von allen Elementen der nächsthöheren Hierarchieebene überdeckt (vgl. Kapitel 2.1.1). Dadurch werden alle Elemente in der Treemap angezeigt, deren Hierarchieebene größer ist als die der Wurzel. Um eine detailliertere Sicht auf die Unterelemente eines bestimmten Elements zu erhalten, muss dieses mit der Maus zuerst ausgewählt werden. Hiermit wird das ausgewählte Element die neue Basis der Treemap. In der Treemap werden dadurch nur die Unterelemente angezeigt, die Bestandteil der neuen Basis sind und deren Hierarchieebene größer ist als die der neuen Wurzel. Um die Basis in eine niedrigere Hierarchieebene zurückzuführen, wurde in der linken Toolbar eine 'breadcrumb navigation' implementiert.

Die Farbgebung der Elemente dieser Oberfläche wurde, wie in Kapitel 2.1.2 beschrieben, umgesetzt. Die Elemente, die Nachfolger enthalten, werden in Grautönen dargestellt. Im Gegensatz dazu werden die Blattknoten mit Zufallsfarben belegt. Die Unterelemente desselben Elements haben die gleiche Farbe. Bei der Farbgebung wurde darauf geachtet, dass möglichst dezente Farbtöne und keine Signalfarben (z.B. Grün, Rot) verwendet wurden. Die unterschiedlichen Farben unterstützen zudem bei der Differenzierung der Elemente. Eine Neuverteilung der Farben kann mit der implementierten Funktion in der unteren Toolbar ausgeführt werden.

Wenn ein Benutzer mit der Maus über eine Fläche fährt, dann wird diese mit einem roten Rahmen markiert. Dabei werden ihm zusätzliche Informationen zu diesem Element in einem Tooltip angezeigt. Die Markierung hält nur solange sich die Maus über der Fläche dieses Elementes befindet. In dem Tooltip werden Informationen aus dem Datensatz angezeigt, wie z.B. der Name des Elements, die Hierarchieebene, in der sich dieses befindet und das Budget, das diesem zugeordnet wurde. Ein Tooltip ist ein kleines Fenster, welches sich meist oberhalb des Mauszeigers befindet und mit diesem verbunden ist.

4.2.2 Datenänderung

Die Datenänderungsoberfläche (Abbildung 15) ist die wichtigste Oberfläche dieser Treemap. Mit deren Hilfe werden die Budgets bei der Strategieplanung verändert oder angepasst. Die Treemap muss daher eine gute Stabilität und Anordnung der Daten aufweisen. In den folgenden Abschnitten wird diese Oberfläche vorgestellt und die Interaktionsmöglichkeiten der Benutzer erklärt.

Um von der Startseite zur Datenänderungsoberfläche zu gelangen, muss diese zuerst mithilfe der Schaltfläche in der linken Toolbar aktiviert werden. Die Darstellung der Datenänderungsoberfläche entspricht weitgehend der Oberfläche der Datenexplorationsebene. Der einzige Unterschied liegt darin, dass jede Fläche eines Elements auf dieser Oberfläche über neue Schalt- und Eingabeflächen verfügt. Diese werden für die Veränderung der Flächengrößen bzw. des Budgets benötigt. Innerhalb der Flächen der Blattknoten wird zusätzlich die Budgethöhe des jeweiligen Elements angezeigt. Ein Benutzer erhält dadurch zur Flächengröße einen Referenzwert. Dies vereinfacht das Ändern oder Vergleichen der Budgethöhe eines Elements.

Ein Benutzer hat zwei Möglichkeiten, eine Budgetänderung durchzuführen: zum einen durch die Eingabe eines neuen Wertes in ein Textfeld und zum anderen durch die Manipulation der Flächengröße mithilfe der Maus. Jedes Element enthält dabei zwei Schaltflächen, mit denen sich die Budgets verändern lassen.

Abbildung 14 verdeutlicht, wie eine Änderung mithilfe dieser Schaltflächen ausgeführt werden kann. Bei der ersten Variante kann eine Fläche bzw. ein Budget durch das Ziehen und Ablegen der Schaltfläche (engl. 'drag and drop') mit der Maus verändert werden.

Die Verteilung des geänderten Budgets, wird durch das Konzept in Kapitel 4.1 beschrieben (vgl. Abbildung 11). Das Budget eines Elements kann außerdem durch die Eingabe eines Wertes verändert werden. Dazu muss die in Abbildung 14.2 dargestellte Schaltfläche betätigt werden. Es erscheint ein Textfeld, in das der Benutzer einen Wert eingeben kann. Anschließend wird mit der Taste "Enter" bestätigt. Bei einem Blattknoten existiert diese Schaltfläche nicht. Hierbei muss das Element bzw. die Fläche zuerst ausgewählt werden. Die Eingabe erfolgt dann äquivalent. Die Änderungen in der Treemap werden in der Chronik gespeichert. Diese befindet sich in der linken Toolbar.

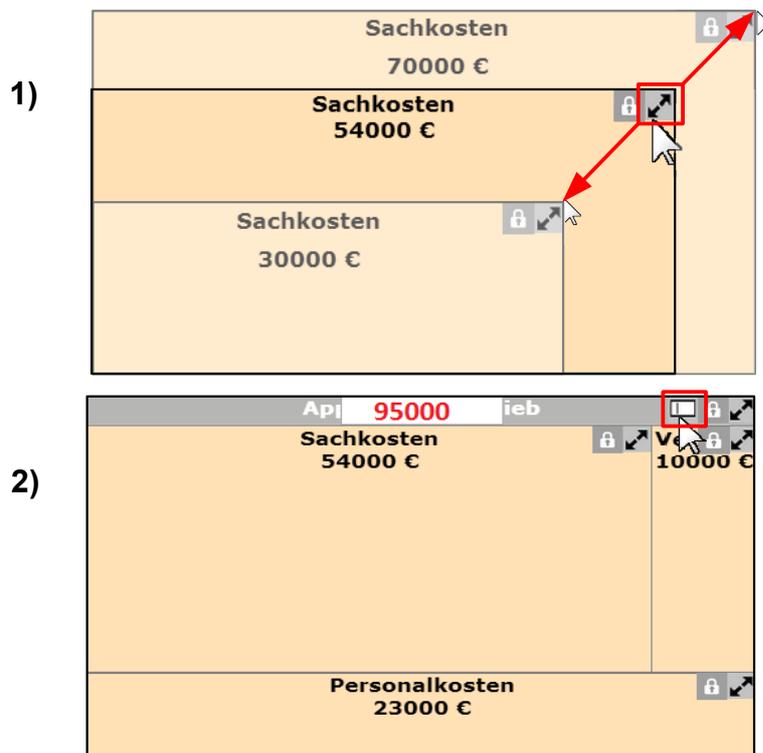


Abbildung 14: Treemap-Implementierung, Änderungsoptionen

Soll das Budget eines Elements gesperrt werden, sodass dieses durch Wertänderungen der anderen Elemente nicht beeinflusst wird, dann kann dies über die integrierte Schaltfläche mit dem Schlosssymbol erfolgen. Diese Schaltfläche kann bei einem Element drei unterschiedliche Zustände einnehmen (Abbildung 16).



Abbildung 16: Schaltflächen zum Sperren eines Budgets

Das graue Schloss beschreibt den Ausgangszustand eines Elements, in dem das Budget nicht gesperrt ist. Der Benutzer kann daher das Budget dieses Elements über dessen Schaltflächen ändern. Außerdem kann das Budget durch Wertänderungen anderer Elemente beeinflusst werden (siehe Kapitel 4.1, Abbildung 12).

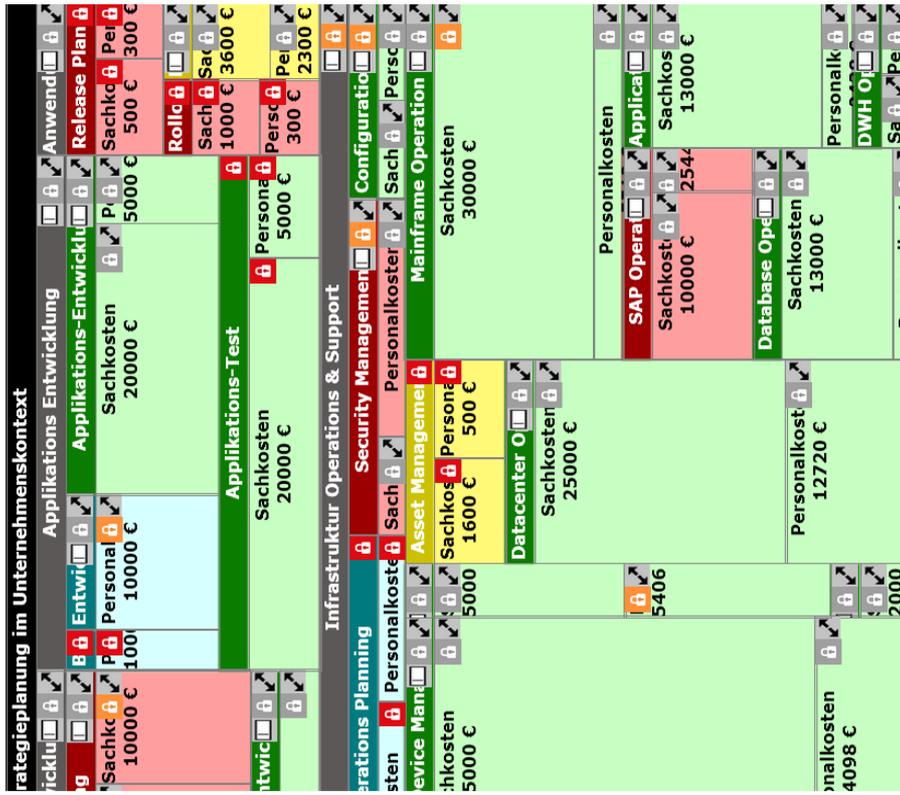
Wird diese Schaltfläche mit der Maus ausgewählt, dann wird das jeweilige Element gesperrt und die Schaltfläche wird Orange. Dadurch kann der Budgetwert des ausgewählten Elements nicht mehr durch den Benutzer in der Treemap verändert werden. Um zum entsperrten Zustand zurückzukehren, muss die orangene Schaltfläche erneut betätigt werden.

Die rote Schaltfläche beschreibt den Zustand eines Elements, bei dem das Budget sich unter keinen Umständen ändern darf. Dabei können diese Elemente in der Treemap nicht verändert oder entsperrt werden. Dies ist nur direkt im Datensatz möglich. Die rote Schaltfläche wurde für das Budget von Unternehmensbereichen konzipiert, die in einem Unternehmen wichtige Funktionen übernehmen. Das Ändern dieser Budgets wäre in diesem Fall mit hohen Verlusten verbunden. Durch die Funktionseinschränkung kann das Budget in diesen Bereichen nicht versehentlich verändert werden.

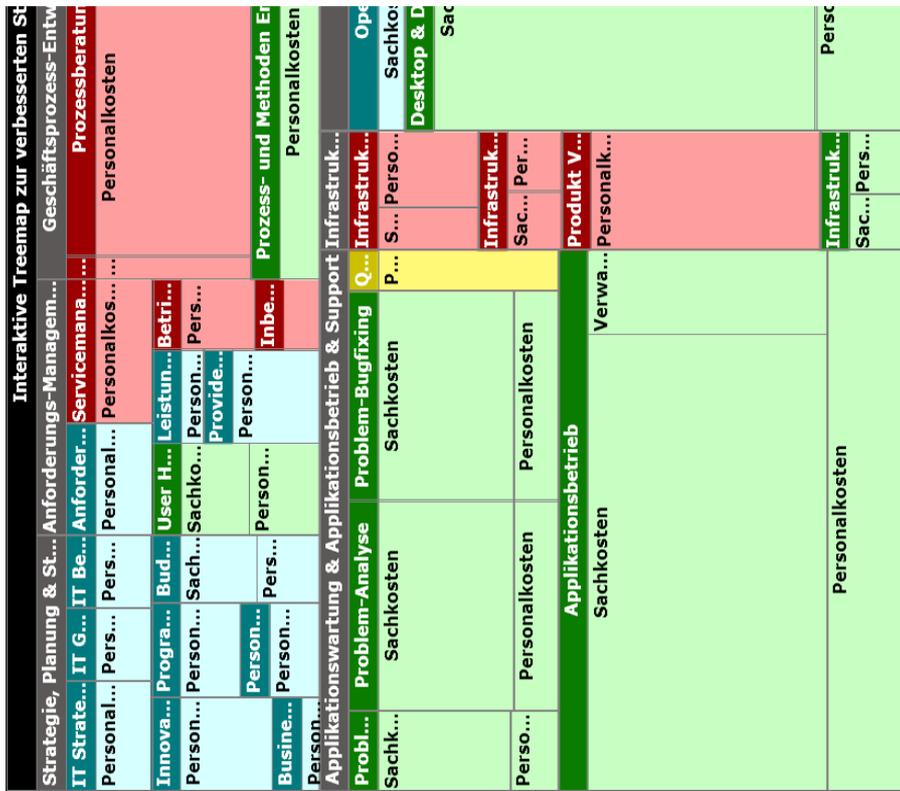
4.2.3 Funktionsklassifikation

Ein Business-Datensatz enthält zusätzlich zum Budget auch noch andere Daten. Diese können bei der Strategieplanung in einem Unternehmen auch von Bedeutung sein. Daher beschreibt die dritte Benutzeroberfläche der Implementierung, die Funktionsklassifikation der Elemente eines Datensatzes (Abbildung 17). Hierbei werden die Eigenschaften und Funktionen der Elemente mithilfe von Farben in der Treemap dargestellt. Elemente mit gleichen Farben verfügen über gleiche Eigenschaften und Funktionen in einem Unternehmen. Diese können direkt aus dem Datensatz ausgelesen werden. Außerdem können sich die Eigenschaften eines Elements im Laufe der Interaktion durch bestimmte Vorgaben im Anwendungscode oder im Datensatz ändern.

Der Datensatz, aus dem die Treemap in dieser Implementierung modelliert wurde, enthält zusätzlich Informationen darüber, ob die einzelnen Aufgabenbereiche der Abteilungen vollständig, teilweise oder nicht an Fremdfirmen vergeben werden dürfen. Im Datensatz wird für diese Funktionsklassifikation eine eigene Spalte verwendet. Die einzelnen Aufgabenbereiche können dabei folgendermaßen bewertet werden: 'non-critical', 'fundamental', 'important' oder 'vital'. Aufgabenbereiche, die als 'non-critical' oder 'fundamental' bewertet wurden, können dabei bedenkenlos an Fremdfirmen vergeben werden. Wenn eine Abteilung in einem Bereich sensible Daten oder geheime Fertigungstechnologien/-verfahren verwendet, dann darf dieser nicht oder nur eingeschränkt fremdvergeben werden. Die Informationen in diesem Bereich sind für das Kernbusiness eines Unternehmens wichtig und stellen so ein Unterscheidungsmerkmal zu anderen Mitbewerbern dar. Im Datensatz werden diese Aufgabenbereiche mit den Bewertungen 'important' oder 'vital' belegt.



mit Datenänderungsoption



ohne Datenänderungsoption

Abbildung 17: Treemap-Implementierung, Funktionsklassifikation

5 Evaluation

Bei dieser Bachelorarbeit wurde nicht nur Wissen im Bereich der Visualisierung und der Softwareentwicklung von interaktiven Systemen vorausgesetzt, sondern auch im Bereich der Wirtschaft und des Unternehmensmanagements. Die Durchführung dieser Bachelorarbeit wurde sowohl von einem wissenschaftlichen Mitarbeiter der Universität Stuttgart als auch von einem Mitarbeiter der Dualen Hochschule Baden-Württemberg (kurz DHBW) betreut. Nach der Entwicklungsphase des Prototyps wurde das Konzept evaluiert. Die Evaluation fand am Institut für Visualisierung und Interaktive Systeme (kurz VIS) der Universität statt. In einer Sitzung mit Mitarbeitern des IT- und Beratungsunternehmens IBM wurde darüber hinaus ermittelt, wie relevant das entwickelte Konzept für die Aufgabengebiete in einem Unternehmen ist. In diesem Kapitel werden das Umfeld, der Ablauf und die Ergebnisse der Evaluation und des Gesprächs vorgestellt.

5.1 Umfeld und Ablauf

Bei der Evaluation an der Universität nahmen 5 Probanden teil. Dabei handelte es sich sowohl um wissenschaftliche Mitarbeiter des VIS Instituts als auch um Absolventen der Universität. Jeder Proband durfte das Konzept anhand des Prototyps auf einem Notebook testen. Der Ablauf der Evaluation verlief in 3 Phasen.

In der ersten Phase wurde die Evaluation geplant. Im Zuge dessen wurden Probanden gesucht, die im Bereich der Visualisierung und Interaktive Systeme erfahren sind. Daher bot es sich an, Mitarbeiter und Absolventen des VIS Institutes der Universität Stuttgart auszuwählen. Des Weiteren wurde ein Raum gesucht, in welchem die Evaluation möglichst ungestört durchgeführt werden konnte. Bei der Evaluation wurden verschiedene Unterlagen, wie z.B. Tabellen, Diagramme und der Prototyp selbst verwendet. Diese mussten davor aufbereitet werden.

In der zweiten Phase fand die Durchführung der Evaluation statt. Der Ablauf verlief folgendermaßen:

1. Vorstellung der Aufgabenstellung und Vermittlung von Kenntnissen im Bereich der Wirtschaft und Visualisierung, die zum Verstehen des Konzepts benötigt werden.
2. Vorstellung des Konzepts anhand des Prototyps.
3. Test des Prototyps durch die Probanden.
4. Bewertung des Konzepts und des Prototyps mithilfe eines Fragebogens.
5. Befragung der Probanden zu dem Konzept und der Interaktion im Prototyp.

Am Anfang der Evaluation wurde jedem Probanden das benötigte Grundwissen im Bereich der Visualisierung mit Treemaps und der Strategieplanung in einem Unternehmen vermittelt. Hier wurden verschiedene Visualisierungsmethoden vorgestellt, die zur Strategieplanung eingesetzt werden können. Diese Methoden wurden anschließend bezüglich ihrer Funktionalität, Komplexität, Übersichtlichkeit und Benutzerfreundlichkeit mit der Treemap des Prototyps verglichen.

Als Nächstes wurde den Probanden das Konzept anhand des Prototyps vorgestellt. Anschließend durften die Probanden das Konzept mithilfe eines Anwendungsfalls (siehe Anhang) testen, durch den diese alle Funktionen und Eigenschaften des Konzepts erproben konnten. Bei dieser Evaluation kamen unterschiedliche Evaluationsmethoden und Hilfsmittel zum Einsatz. So wurde unter anderem ein Fragebogen verwendet, der eine gezielte Rückmeldung zur Funktionalität und Darstellung des Konzepts lieferte (siehe Kapitel 5.2.2). Darüber hinaus wurde die Methode des 'Lauten Denkens' genutzt, bei der die Probanden aufgefordert werden, ihre Gedanken während der Interaktion mit dem Prototyp laut auszusprechen. Nach dem Test und der Bearbeitung des Fragebogens, fand eine Befragung der Probanden zum Konzept und der Visualisierung statt.

Die dritten und letzten Phase der Evaluation beschrieb die Auswertung der Ergebnisse.

5.2 Ergebnisse

In den folgenden Abschnitten werden die Ergebnisse der Evaluation vorgestellt.

5.2.1 Beobachtung

Durch die Evaluationsmethode des 'Lauten Denkens', konnte der Gedankengang der Probanden in Form von Notizen festgehalten werden. Dadurch konnten die Bereiche des Konzepts und der Darstellung identifiziert werden, bei denen Interaktions- oder Verständnisschwierigkeiten auftraten. Die Ergebnisse dieser Evaluationsmethode und die Beobachtungen während des Tests werden in folgenden Punkten beschrieben.

Alle Probanden ...

- waren in der Lage, nach einer kurzen Einführung und Erklärung des Konzepts, dieses anhand des Prototyps eigenständig zu testen.
- konnten auf Fragen bzgl. der Funktionalität des Konzepts, eine richtige Antwort liefern.
- waren der Ansicht, dass die Interaktion mit der Schaltfläche und dem Textfeld einfacher, präziser und besser als die Interaktion mittels 'drag and drop' ist.
- fanden das Konzept und die Interaktion innovativ.
- fanden die Idee hinter dem Konzept gut und für diesen Aufgabenbereich angemessen.
- haben die Vorteile der Strategieplanung mithilfe dieses Konzepts und der Visualisierung erkannt.
- haben ein ähnliches Konzept, in einer Treemap noch nie gesehen.
- fanden den Prototypen gut und hatten Gefallen daran, diesen auszuprobieren.

Vier Probanden ...

- fanden das Konzept leicht verständlich und fühlten sich in der Lage, dieses mithilfe des Prototyps auch ohne fremde Hilfe innerhalb kürzester Zeit zu verstehen und anzuwenden.
- bevorzugten dieses Konzept und die Visualisierungsform für diesen Aufgabenbereich.
- fanden die Umsetzung des Sperrens eines Budgets, mittels der angebotenen Schlossschaltfläche gut.

Drei Probanden ...

- fanden die Interaktion mittels 'drag and drop' gewöhnungsbedürftig.
- hatten Probleme bei der Interaktion mittels 'drag and drop'. Hierbei war unter anderem unklar, in welche Richtung die Maus bewegt werden muss, um die Größe eines Elements zu verändern.
- hatten Schwierigkeiten, die Funktion der Schaltflächen aus deren Darstellung zu erkennen.
- vermissten ein Feedback bei einer nicht ausführbaren Interaktion.

Ein Proband ...

- fand das Konzept sehr komplex. Dabei meinte er, dass dieses ohne vorherige Erklärungen oder Tutorial, nicht so einfach zu verstehen wäre.
- würde bei der Strategieplanung zusätzlich zu der Treemap, auch eine Tabelle nutzen. Diese soll zur Eingabe der Wertänderungen dienen.
- fand die Änderungsoberfläche aufgrund der vielen Schaltflächen unübersichtlich.

5.2.2 Fragebogen

Der verwendete Fragebogen bezieht sich auf die sieben Grundsätze der Dialoggestaltung, die in der ISO-Norm 9241 Teil 110 festgelegt sind [23]. Sowohl die Fragen als auch die Kategorien wurden zum Teil leicht abgeändert und an diese Evaluation angepasst. Von den sieben Kategorien der ISO-Norm wurden nur drei Kategorien im Fragebogen verwendet.

Die restlichen Kategorien waren für die Evaluation des Konzepts unbedeutend. Jede Kategorie besteht aus mehreren Aussagen, die von den Probanden als 'teilweise zutreffend', 'zutreffend', 'teilweise nicht zutreffend', 'nicht zutreffend' oder 'neutral' bewertet werden konnten.

Nachfolgend werden die Ergebnisse in den einzelnen Kategorien vorgestellt, die durch diese Evaluationsmethode erzielt wurden. Bei jeder Aussage werden die erzielten Bewertungen aller Probanden angegeben.

Seitenverhältnisse, Stabilität, Anordnung	
Sorgen die Seitenverhältnisse, die Stabilität und die Anordnung der Elemente einer Treemap bei der Interaktion für eine übersichtliche Visualisierung?	
Die Flächenvolumen der Elemente in der Treemap lassen sich gut miteinander vergleichen.	2 Probanden: trifft zu 3 Probanden: trifft teilweise zu
Die Seitenverhältnisse der Flächen sind gut.	5 Probanden: trifft teilweise zu
Bei der Veränderung der Flächengröße eines Elements, bleibt dieses weitgehend auf seiner Ausgangsposition.	3 Probanden: trifft zu 2 Probanden: trifft teilweise zu
Bei der Veränderung der Flächengröße eines Elements, bleiben die anderen Elemente weitgehend auf ihrer Ausgangsposition.	3 Probanden: trifft zu 2 Probanden: trifft teilweise zu
Anmerkungen (optional):	
- Bei zu kleinen Flächen wird die Schrift unleserlich	

Aufgabenangemessenheit	
Unterstützt dieses Konzept den Benutzer bei der Strategieplanung in einem Unternehmen ohne ihn unnötig zu belasten? Das Konzept ...	
ist unkompliziert zu bedienen.	1 Proband: trifft zu 3 Probanden: trifft teilweise zu 1 Proband: neutral
eignet sich gut um die anfallenden Aufgaben effizient zu bewältigen.	3 Probanden: trifft zu 2 Probanden: trifft teilweise zu
lässt sich auch in anderen Aufgabenbereichen einsetzen.	4 Probanden: trifft teilweise zu 1 Proband: neutral
und die gebotenen Interaktionsmöglichkeiten sind in diesem Aufgabenumfeld angemessen.	1 Proband: trifft zu 4 Probanden: trifft teilweise zu

Erwartungskonformität, Interaktion, GUI	
Kommt dieses Konzept durch eine einheitliche und verständliche Gestaltung Ihren Erwartungen und Gewohnheiten entgegen?	
Das User Interface ist übersichtlich gestaltet.	3 Probanden: trifft teilweise zu 2 Probanden: neutral
Die Funktionsweise des Konzepts und der Interaktion ist leicht verständlich.	2 Probanden: trifft zu 2 Probanden: trifft teilweise zu 1 Proband: neutral
Die Interaktionsmöglichkeiten sind klar erkennbar und verständlich.	1 Proband: trifft zu 2 Probanden: trifft teilweise zu 2 Probanden: neutral
Anmerkungen (optional):	
<ul style="list-style-type: none"> - Probleme bei Interaktion mit 'drag and drop' - Gute Icons für Bearbeiten von Abteilungen - Farbgebung unklar - Chronik unklar - Rückmeldung im Fehlerfall fehlt 	

Erwartungskonformität, Interaktion, GUI	
Kommt dieses Konzept durch eine einheitliche und verständliche Gestaltung Ihren Erwartungen und Gewohnheiten entgegen? Das Konzept ...	
erleichtert die Orientierung, durch eine einheitliche Gestaltung.	3 Probanden: trifft zu 2 Probanden: trifft teilweise zu
gibt ein klares Feedback/Rückmeldung ob eine Aktion erfolgreich war.	3 Probanden: trifft zu 2 Probanden: trifft teilweise zu
lässt sich durchgehend nach einem einheitlichen Prinzip bedienen.	4 Probanden: trifft zu 1 Proband: trifft teilweise zu

5.2.3 Meinungsumfrage

Die Probanden wurden abschließend zu dem getesteten Konzept befragt. Dabei wurden folgende Fragestellungen verwendet:

- 1) Wie gut sind Sie mit den angebotenen Interaktionsmöglichkeiten bzw. -schaltflächen zurechtgekommen?
- 2) War das Konzept für Sie leicht verständlich? Haben Sie verstanden, wie sich eine Erhöhung des Budgets eines Elements auf die anderen Budgets auswirkt? Haben Sie das Konzept des Sperrens eines Budgets verstanden? Bitte geben Sie die Funktionsweise des Konzepts in eigenen Worten wieder.
- 3) Hätten Sie das Konzept auch ohne vorherige Einführung verstanden?
- 4) Wenn Ihnen bei der Strategieplanung eine Tabelle, ein Graph oder eine Treemap zur Verfügung stehen würde, welche dieser Visualisierungen würden sie bevorzugen/verwenden?

- 5) Haben Sie Verbesserungsvorschläge bzgl. des Konzepts, des Designs der Oberfläche oder der verwendeten Visualisierungsform?

Viele dieser Fragen wurden bereits durch die Beobachtung der Probanden während des Tests und der Evaluationsmethode des 'Lauten Denkens' beantwortet. Dadurch wurden die Erkenntnisse aus dem Kapitel 5.2.1 nochmals bestätigt. Zudem wurden einige Verbesserungsvorschläge genannt, die in Kapitel 6 vorgestellt werden.

5.2.4 Sitzung

Nach der Evaluation wurde ermittelt, ob das Konzept im realen Umfeld eines Unternehmens eingesetzt werden kann. Dazu wurde das Konzept in einer Sitzung zwei Mitarbeitern des Unternehmens IBM vorgestellt. Die Resonanz dieser Sitzung war positiv.

Im Folgenden werden drei Punkte aufgeführt, die das Ergebnis dieser Sitzung wiedergeben:

- Es ist realistisch, dass das Endprodukt dieses Entwurfs in einem Unternehmen zur Strategieplanung eingesetzt werden kann und diese unterstützt.
- Treemaps wurden bis jetzt nur selten in einem Unternehmen eingesetzt, da ein konkreter Anwendungsfall und eine passende Realisierung fehlten.
- Das Konzept ist ausbaufähig und sollte weiterentwickelt werden. Im jetzigen Konzept fehlen unter anderem die Auswirkungen, die durch eine Budgetänderung erfolgen (durch eine Budgetkürzung müssen eventuell Mitarbeiter des Unternehmens entlassen werden).

6 Fazit und Ausblick

Ziel dieser Bachelorarbeit war es, ein Konzept zur Strategieplanung in einem Unternehmen zu erstellen und dieses in eine Visualisierung zu integrieren. Besonders großen Wert wurde auf eine anschauliche Darstellung und eine einfache Datenexploration und Datenmanipulation gelegt. Das Konzept beschreibt, wie sich durch die Veränderung des Budgets eines Unternehmensbereichs die Budgets der anderen Unternehmensbereiche in einer Treemap ändern. Zusätzlich zum Konzept wurde ein funktionsfähiger Prototyp erstellt, anhand dem das Konzept evaluiert werden konnte. Nachfolgend werden die Erkenntnisse beschrieben, die aus der Evaluation resultieren. Diese sollen bei zukünftigen Implementierungen die Entwickler dabei unterstützen, sowohl das Konzept als auch die Visualisierung zu verbessern.

Die Auswertung der Ergebnisse zeigt, dass sich die Grundidee des Konzepts sehr gut eignet, um die anfallenden Aufgaben in dem vorgesehenen Einsatzbereich effizient zu lösen. Das Konzept erleichtert dem Benutzer nicht nur die Arbeit mit großen Datensätzen, sondern stellt bei Veränderungen die Zusammenhänge der Daten anschaulich dar. Es ist zudem einfach verständlich und kann nach einer kurzen Einführung schnell erlernt werden. Das Konzept ist ebenso für andere Aufgabenbereichen geeignet. Ein Beispiel dafür ist die Verwaltung und Anpassung der Gehälter von Mitarbeitern eines Unternehmens. So kann z.B. eine Gehaltserhöhung, auf alle Mitarbeiter einer Abteilung gleichverteilt werden.

Die einzige Schwachstelle liegt bei der Interaktion in der Treemap. Viele der Probanden konnten die vorgegebenen Schaltflächen nicht ihrer Funktion zuweisen. Daher ist es notwendig nur solche Schaltflächen zu verwenden, die die Benutzer aus anderen Systemen oder Anwendungen kennen.

Weiterhin zeigten sich Schwierigkeiten bei der Interaktion mittels 'drag and drop'. Für viele Probanden erschien diese Interaktionsmöglichkeit gewöhnungsbedürftig. Dabei hatten diese vor allem Probleme bei der Veränderung eines Budgets. Denn die Flächengröße einer ausgewählten Fläche verändert sich bei einem Squarified-Algorithmus nicht in die Richtung, in die die Maus bei der Interaktion bewegt wird. Dies liegt vor allem daran, dass der verwendete Algorithmus die Flächen immer in die Richtung positioniert, aus welcher sich die besten Seitenverhältnisse ergeben. Ein Verbesserungsvorschlag wäre daher, die Flächengrößen nicht über eine 'drag and drop' Schaltfläche zu verändern, sondern durch das Verziehen der Seitenkanten eines Elements in die gewünschte Richtung (Abbildung 18).

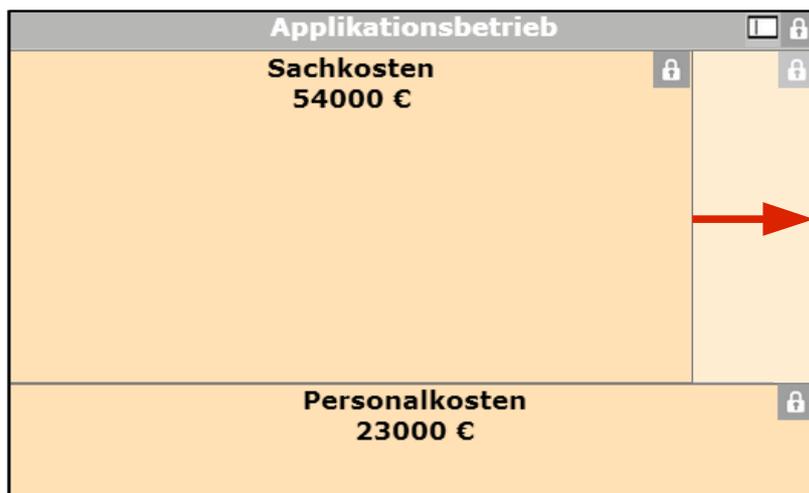


Abbildung 18: Verbesserungsvorschlag 1

Die Elemente der Treemap sollen erst dann neu angeordnet werden, wenn der Benutzer mit der Veränderung fertig ist. Im Gegensatz dazu verlief die Interaktion, bei der der Benutzer die Wertänderung in ein Textfeld eingibt, ohne Probleme. Dabei muss beachtet werden, dass spätere Implementierungen eventuell auf mobilen Geräten mit Touch-Interaktion ausgeführt werden. In diesem Fall ist die Interaktion mittels der Eingabefläche genauer und einfacher zu bedienen.

Einigen Probanden bemängelten, dass die Flächen kleiner Elemente bei der Änderungsoberfläche nicht sichtbar sind. Dadurch werden die Bezeichnungen und die Angaben zum Budget innerhalb der Flächen überdeckt. Dieses Verhalten kann vermieden werden, indem z.B. nur die Interaktionsschaltflächen von dem Element angezeigt werden, das man verändern möchte. Dabei muss das Element zuvor ausgewählt werden (Abbildung 19). Die Treemap wird so übersichtlicher.

Um das Finden eines Knotens in einer Treemap zu erleichtern sollte eine Suchfunktion implementiert werden. Zusätzlich dazu soll ein Benutzer ein klares Feedback erhalten, wenn eine Budgetänderung aufgrund eines gesperrten Budgets nicht möglich ist.

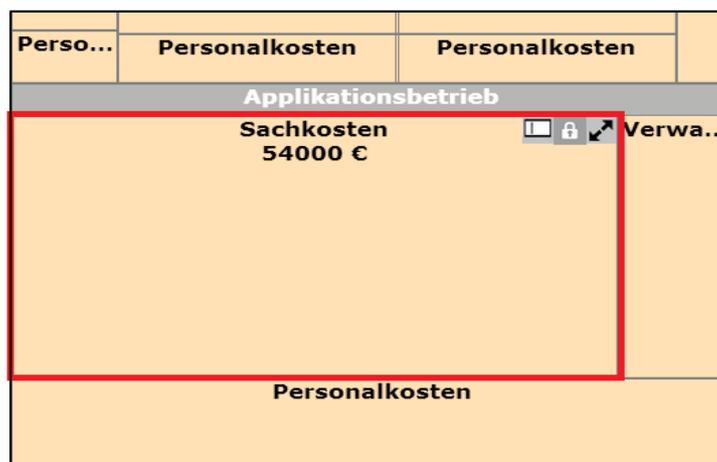


Abbildung 19: Verbesserungsvorschlag 2

Abschließend ist anzumerken, dass das Gesamtkonzept eingeschlossen der Visualisierung, einen insgesamt guten Eindruck bei den Probanden hinterlassen hat. Alle Probanden waren der Ansicht, dass das Konzept und die Visualisierung zu einem Endprodukt weiterentwickelt werden sollten. Diese Aussage wurde auch von den Mitarbeitern des Unternehmens IBM vertreten.

Trotz der positiven Resonanz dieser Bachelorarbeit setzen Unternehmen heutzutage, im Vergleich zu anderen Visualisierungsformen, Treemaps wenig ein [2]. Der Grund dafür liegt zum einen darin, dass die Benutzer aus Gewohnheit alt bewährte Visualisierungsmethoden bevorzugen. Zum anderen fehlen in diesem Bereich oft brauchbare Softwarelösungen, die auf die Anwendungsgebiete der Benutzer zugeschnitten sind. Dennoch werden Treemaps zukünftig noch stets weiterentwickelt und erschließen somit neue Einsatzgebiete. Ein Beispiel dafür stellt die klassische Squarified Treemap-Form dar, die sich im Laufe der Zeit vielfältig abgewandelt und neuen Aufgabenbereichen angepasst hat. Dadurch entstehen viele neue Treemap-Algorithmen, die z.B. Treemaps mit runden oder dreidimensionalen Flächen [2] erstellen. Somit wird eine immer detailliertere Sicht auf die Daten gewährleistet.

In Unternehmen können Treemaps in vielen Bereichen eingesetzt werden. Durch ihre platzsparende Struktur und ihrer übersichtlichen Darstellung unterstützen Treemaps den Benutzer sowohl bei der Datenanalyse als auch bei der Datenmanipulation. Dementsprechend werden mittlerweile immer mehr Treemap-Anwendungen von Softwareunternehmen für diesen Bereich entwickelt.

Nachfolgend werden drei dieser Anwendungen aufgelistet:

- Treemap Business Data Visualization (Abbildung 20) [24]
- Salesforce Management Treemap [21]
- Business Analysis Tool Desktop (Abbildung 21) [25]

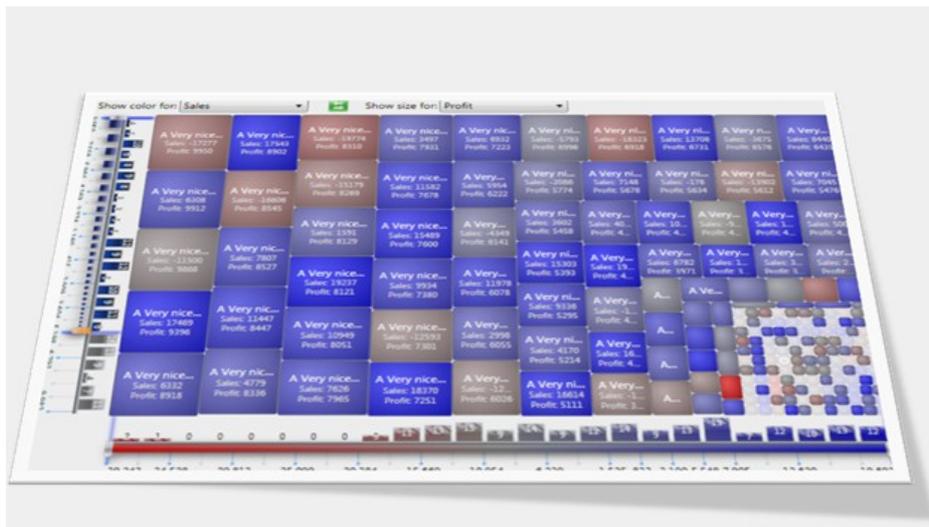


Abbildung 20: Treemap Business Data Visualization [24]

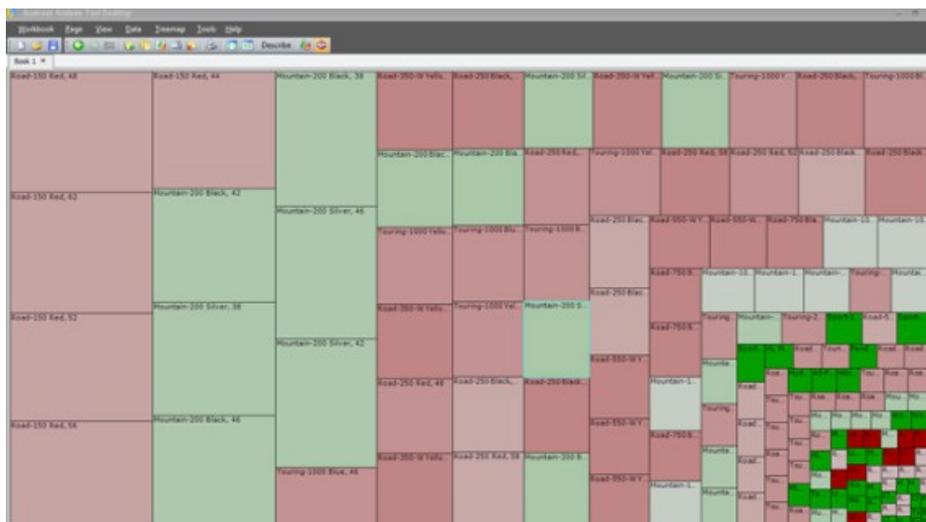


Abbildung 21: Business Analysis Tool Desktop [25]

Diese Visualisierungen wurden zur Darstellung der Unternehmensstruktur, Verwaltung und Überwachung einzelner Bereiche eines Unternehmens entwickelt. Bei allen Implementierungen steht die Datenexploration im Vordergrund. Eine direkte Datenmanipulation in der Treemap ist jedoch nicht möglich [21] [24] [25].

In den meisten Einsatzbereichen einer Treemap fehlen ein passendes Konzept und eine funktionsfähige Implementierung. Mit der steigenden Anzahl an Informationen, die täglich in Unternehmen anfallen, wird es jedoch in Zukunft unabdingbar werden, eine übersichtliche und auch effiziente Visualisierungsmethode zu finden, zu entwickeln und einzusetzen. Genau in diesem Bereich kann eine Treemap gegenüber anderen Visualisierungen überzeugen.

Literaturverzeichnis

- [1] Frankfurter Societäts-Medien GmbH: Unvorstellbare Datenmengen (2013). <http://www.deutschland.de/de/node/2828>
- [2] Ben Shneiderman: "Treemaps for space-constrained visualization of hierarchies" (2009). <http://www.cs.umd.edu/hcil/treemap-history/>
- [3] Christian Grlica: "Visualisierung im Information Retrieval System". *Möglichkeiten und Methoden zur Informationsvisualisierung in einem Information Retrieval System* (2002); Grin Verlag, Norderstedt
- [4] IBM: "Visualizations : Export wichtiger Waren. Treemap" (2010). <http://www-958.ibm.com/software/analytics/manyeyes/visualizations/export-wichtiger-waren-treemap>
- [5] MarketWatch: "Map of the Market" (2013). www.marketwatch.com/tools/stockresearch/marketmap
- [6] Vogel Business Media: "Datenstruktur" (2013). <http://www.process.vogel.de/index.cfm?pid=2995&title=Datenstruktur>
- [7] Rob Orsini: "Treemap on Rails" (2006). http://www.oreillynet.com/ruby/blog/2006/07/treemap_on_rails.html
- [8] TIBCO Software: "What is a Treemap?" (2013). http://stn.spotfire.com/spotfire_client_help/tree/tree_what_is_a_treemap.htm
- [9] Thorsten Foerstemann: "Analyse von Servicedaten medizin-technischer Geräte in Krankenhäusern" (2004). <http://www.foerstemann.name/misc/doktorarbeit/doktorarbeit.html>, Medizinische Hochschule Hannover, Dissertation.
- [10] Drew Skau: "The Giving Tree(map)" (2012). <http://blog.visual.ly/the-giving-treemap/>
- [11] Mark Bruls; Kees Huizing; Jarke J. van Wijk: "Squarified Treemaps" (2013). <http://www.win.tue.nl/~vanwijk/stm.pdf>, Technische Universität Eindhoven
- [12] Björn Engdahl: "Ordered and Unordered Treemap Algorithms and Their Application on Handheld Devices" (2005). http://www.nada.kth.se/utbildning/grukth/exjobb/rapportlistor/2005/rapporter05/engdahl_bjorn_05033.pdf, Royal Institute of Technology, Sweden, Masterthesis
- [13] Nicolas Garcia Belmonte: "DOM vs. Canvas TreeMaps" (2013). <http://philogb.github.io/blog/2010/01/24/dom-vs-canvas-treemaps/>
- [14] Holger Gronski: "Architekten und Ingenieure" (2013). <http://www.g1architektur.de/html/kostenmanagement.html>

- [15] Bernd Baumgarten; Christoph Neumann; Matthias Scheffler : “Geschäftsmodelle auf Basis des Open Source Betriebssystems LINUX“ (2011).
http://winwiki.wi-fom.de/index.php/Geschäftsmodelle_auf_Basis_des_Open_Source_Betriebssystems_LINUX
- [16] Peter Posluschny; Myra Posluschny: “Trainingsbuch Controlling“ (2006). Rudolf Haufe Verlag GmbH & Co. KG, München
- [17] IBM: “Listing Visualizations“ (2013). <http://www-958.ibm.com/software/analytics/manyeyes/visualizations>
- [18] Rolf Mohr: “Skript zum Thema: Rechnungswesen / Controlling“ (1. Auflage, 2013). Grin Verlag, Norderstedt
- [19] IBM: “Visualizations Business treemap“ (2012). <http://www-958.ibm.com/software/analytics/manyeyes/visualizations/business-treemap>
- [20] Roel Vliegen: “Visualizing Business Information using Generalized Treemaps“ (2006). <http://alexandria.tue.nl/extra2/afstversl/wsk-i/vliegen2006.pdf>, Technische Universität Eindhoven, Masterthesis
- [21] Ben Shneiderman: “Discovering Business Intelligence Using Treemap Visualizations“ (2006). <http://www.perceptualedge.com/articles/b-eye/treemaps.pdf>
- [22] Nicolas Garcia Belmonte: “JavaScript InfoVis Toolkit“ (2013).
<http://philogb.github.io/jit/>
- [23] HTW Chur: “ISONorm 10/110-S“ (2012). <http://www.cheval-lab.ch/was-ist-usability/usabilitymethoden/frageboegen/isonorm-10110-s/>
- [24] Christian Abeln: “Announcement: Treemap Business Data Visualization available on PartnerSource and CustomerSource“ (2009). <http://blogs.msdn.com/b/cabeln/archive/2009/12/05/announcement-treemap-business-data-visualization-available-on-partnersource-and-customersource.aspx>
- [25] Chris Webb: “Business Analysis Tool Desktop“ (2010).
<http://cwebbbi.wordpress.com/2010/01/19/business-analysis-tool-desktop/>
- [26] Benjamin B. Bederson; Ben Shneiderman: “The Craft of INFORMATION VISUALIZATION“ (5. Auflage, 2003). *Reading and Reflections*; Morgan Kaufmann Verlag, San Francisco

Alle URLs wurden zuletzt am 12.11.2013 geprüft.

Anhang

Anwendungsfall:

Alle nachfolgenden Aktionen werden in der Abteilung 'Infrastruktur und Entwicklung' ausgeführt.



(1.1) Erhöhen Sie das Budget der Abteilung 'Infrastruktur und Entwicklung' auf **200 000€**. Verwenden Sie die **Schaltfläche** und das **Textfeld**.



(1.2) Verringern Sie das Budget des Aufgabenbereichs 'Infrastruktur Roadmap' auf **30 000€**. Verwenden Sie die **Schaltfläche** und das **Textfeld**.



(1.3) Erhöhen Sie das Budget des Aufgabenbereiches 'Infrastruktur Architektur & Design' auf **100 000€**. Verwenden Sie die **Schaltfläche** und das **Textfeld**.

(1.4) Klicken Sie auf die Überschrift der **Abteilung 'Infrastruktur und Entwicklung'**



(1.5) Erhöhen Sie die **Sachkosten** des **Aufgabenbereichs 'Infrastruktur und Entwicklung'** **soweit wie möglich**. Führen Sie die Änderung mittels 'drag and drop' aus.



(1.6) Verringern Sie die **Sachkosten** des Aufgabenbereichs '**Infrastruktur Architektur und Design**' **soweit wie möglich**. Führen Sie die Änderung mittels 'drag and drop' aus.



Sperren Sie anschließend das Budget der **Personalkosten** im gleichen Aufgabenbereich.

(1.7) **Navigieren** sie mit der Breadcrumb Navigation zurück zum **Startscreen**.



(1.8) Verringern Sie das Budget der Abteilung 'Infrastruktur und Entwicklung' auf **150 000€**. Verwenden Sie die **Schaltfläche** und das **Textfeld**.

(1.9) Vergewissern Sie sich, dass sich das Budget des Aufgabenbereichs '**Infrastruktur Architektur & Design**' **nicht verändert** hat. (100 000€)

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben.

Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet.

Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens.

Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht.

Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Unterschrift:

Stuttgart, 13.11.2013

Declaration

I hereby declare that the work presented in this thesis is entirely my own.

I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations.

Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before.

The electronic copy is consistent with all submitted copies.

Signature:

Stuttgart, 13.11.2013