

Diplomarbeit

geolyzer — Konzept, Entwicklung und Integration der Usability-Komponenten

Bremen, den 11.10.08

Vorgelegt von

Dirk Köster

Matrikelnummer 146349

unter der Betreuung von

Prof. Dr. Thorsten Teschke

Hochschule Bremen

sowie

Dr.-Ing. Volker Wittpahl

TZI Bremen

im Studiengang

Medieninformatik

der Fakultät 4 (Elektrotechnik und Informatik)
an der Hochschule Bremen

Dieses Diplomarbeit ist unter einem **Creative Commons Namensnennung-Weitergabe unter gleichen Bedingungen 2.0 Deutschland Lizenzvertrag** lizenziert. Der Inhalt darf vervielfältigt, verbreitet und öffentlich aufgeführt und es dürfen Bearbeitungen zu den folgenden Bedingungen angefertigt werden:

Namensnennung: Der Name des Autors muss genannt werden.

Weitergabe unter gleichen Bedingungen: Wird der Inhalt bearbeitet oder in anderer Weise umgestaltet, verändert oder als Grundlage für einen anderen Inhalt verwendet, dann darf der neu entstandene Inhalt nur unter Verwendung identischer Lizenzbedingungen weitergeben werden.

geolyzer – Werkzeug zur automatischen Evaluation der Usability in Google Earth

©2008, Alle Rechte vorbehalten



geolyzer ist unter einem **Creative Commons Namensnennung-Keine kommerzielle Nutzung-Keine Bearbeitung 2.0 Deutschland Lizenzvertrag** lizenziert. Um die Lizenz anzusehen, gehen Sie bitte zu <http://creativecommons.org/licenses/by-nc-nd/2.0/de/> oder schicken Sie einen Brief an **Creative Commons**, 171 Second Street, Suite 300, San Francisco, California 94105, USA.



Diese Diplomarbeit wurde gesetzt aus der *Pigiarniq* und der *Weidemann Serif*. Sie ist zusammen mit der aktuellen Version von **geolyzer** unter <http://www.dirkoester.de> verfügbar.

Dirk Köster

geolyzer — Konzept, Entwicklung und Integration der Usability-Komponenten

Diplomarbeit

Abstrakt

Die vorliegende Arbeit widmet sich der rechnergestützten Verbesserung der Gebrauchstauglichkeit von Geobrowser-Inhalten. Vielen aktuell veröffentlichten Geobrowser-Inhalten fehlt es an einem passenden Usability-Konzept, so dass von einer nutzerfreundlichen Aufbereitung meist nicht gesprochen werden kann. Bislang existierte keine zugeschnittene Methodik und ebenso wenig geeignete Werkzeuge, um Inhalte im Geobrowser zu untersuchen und gegebenenfalls zu verbessern. Einziger Ansatz war bis dato ein Prototyp namens **geolyzer** als Ergebnis einer an der Hochschule Bremen durchgeführten Studie.

Um sich ein entsprechendes Bild der Lage zu machen, wird der Trend zum georeferenziellen Web erläutert und das nötige Hintergrundwissen zu Geobrowsern vermittelt. Es werden gängige Praktiken und Methoden aus dem Usability-Bereich analysiert und auf deren Basis ein passendes Konzept erstellt, um den Prototypen **geolyzer** um Komponenten zur Analyse und Evaluation der Gebrauchstauglichkeit von Geobrowser-Inhalten zu vervollständigen.

Als Ergebnis dieser Arbeit wird aus dem Prototypen ein voll funktionsfähiges Werkzeug erschaffen, mit dem sich Usability-Tests für den Geobrowser Google Earth flexibel, ressourcenschonend und kostengünstig durchführen lassen. Während eines Tests wird jegliche relevante Nutzeraktivität aufgezeichnet und in einer Datenbank abgelegt. Aufgenommene Daten lassen sich visuell aufbereitet exportieren und in Google Earth z.B. als dreidimensionale Heatmap begutachten. Des Weiteren lassen sich PDF-Reports auf Basis des Common Industry Formats automatisch generieren, die neben der reinen Wiedergabe relevanter Aufzeichnungsdaten auch die Gebrauchstauglichkeit der Inhalte automatisch bewerten.

Zur Evaluation der Praxistauglichkeit wird das Werkzeug **geolyzer** bei potentiellen Interessenten ausführlich getestet, um die gesetzten Ziele mit den erhaltenen Resultaten zu konfrontieren.

Inhalt

	Vorwort	8
	Eidesstattliche Erklärung	9
1.	Einführung	10
1.1	Problemdarstellung.	11
1.2	Zielsetzung.	12
1.3	Vorgehen	13
2.	Georeferenzielles Web.	14
2.1	Trend zum Geobrowser	15
2.2	Überblick Earth Viewer	18
2.2.1	Google Earth.	18
2.2.2	NASA Worldwind.	20
2.2.3	ESRI ArcGIS	20
2.3	Georeferenzielles System	21
2.3.1	Geoextraktion	22
2.3.2	Geokodierung	22
2.3.3	Georeferenzielles Modell	23
2.3.4	Keyhole Markup Language	25
2.4	Geobrowser und Usability	33

3.	Usability	38
3.1	Gebrauchstauglichkeit von Produkten	39
3.2	Zugänglichkeit von Web-Inhalten	41
3.3	Qualitätsmerkmale von Inhalten	43
3.4	Untersuchungsverfahren für Inhalte	46
3.5	Werkzeuge für Web-Inhalte	49
3.6	Ansätze für Geobrowser	51
4.	geolyzer als erster Ansatz	53
4.1	Historie	54
4.2	Funktionalität des Ist-Zustandes	55
4.3	Technologie	56
4.3.1	ADO.NET 2.0	57
4.3.2	Transact-SQL	59
4.3.3	Benutzersteuerelemente in .NET	60
4.4	Softwarearchitektur	62
4.5	Soll-Zustand des Endprodukts	63
5.	Entwurf und Realisierung	65
5.1	Konzept der Usability-Komponenten	66
5.1.1	Konzept geolyzer – Quantitativ	66
5.1.2	Konzept geolyzer – Heuristisch	72
5.1.3	Konzept geolyzer – Auto-Evaluierend	75
5.2	Implementierung und Integration	82
5.2.1	Realisierung geolyzer – Quantitativ	82
5.2.2	Realisierung geolyzer – Heuristisch	87
5.2.3	Realisierung geolyzer – Auto-Evaluierend	90
5.3	Testsdurchläufe unter Realbedingungen	94
5.3.1	Nutzertest und Evaluation <i>nuterra</i>	94
5.3.2	Nutzertest visuamundo GbR	96

6.	Diskussion	98
6.1	Evaluation geolyzer – Tracking	99
6.2	Evaluation geolyzer – Heuristisch.	101
6.3	Evaluation geolyzer – Auto-Evaluierend.	102
7.	Zusammenfassung und Ausblick	104
A	Anhang	108
A.1	Datenschema geolyzer.	109
A.2	Klassendiagramm geolyzer	110
A.3	Screenshot geolyzer	111
A.4	Exemplarischer Probandenreport	112
A.5	Exemplarischer Heuristik Report	113
A.6	Exemplarischer geoReport	114
A.7	Ergebnis nuterra Usability-Fragebogen.	115
A.8	Mängel und bekannte Probleme	117
A.9	Wünsche an künftige Versionen.	118
	Quellen	119
	Abbildungen	123
	Tabellen	124
	Abkürzungen	125
	Index	126

*“I don’t particularly like it when people put words
in my mouth, either — unless I say it.”*

George Walker Bush
43rd President of the United States

Vorwort

Als mir und einem Kommilitonen im Januar 2008 die Idee kam, für den Kurs »Webpräsenzen in Geobrowsern« als Projektarbeit den Prototypen eines automatischen Werkzeugs zur Analyse und Verbesserung der Usability im Geobrowser zu entwickeln, war uns nicht bewusst, welche Hürden noch folgen und wie groß der Aufwand letztendlich sein wird. Das Projektziel war klar definiert: innerhalb von zwei Wochen galt es, eine Software zu entwickeln, die als minimaler Prototyp demonstriert, dass eine Datenbank basierte Lösung als Alternative zu herkömmlichen Usability-Verfahren einen neuen, viel versprechenden Ansatz stellt, um gezielt die Usability im Geobrowser zu verbessern. Dieser Prototyp wurde fristgerecht entwickelt. Damit war der erste Stein des Vorhabens gelegt und mehrere Monate weiterer Planung und Umsetzung sollten folgen.

Im Folgekurs »Applikationsentwicklung in Geobrowsern« sollte der Prototyp so erweitert werden, dass bis zum Juli eine marktfähige Software entsteht, in die sich später die Hauptkomponenten – d.h. alle Algorithmen zur Analyse und Verbesserung der Usability – integrieren lassen. Zu dieser Zeit wurde die Projektgruppe um eine Gestalterin ergänzt, die uns die lästige Designlast abnahm und durch ihr Usability-Vorwissen vermittelte, dass der erste Prototyp deutliche Mängel in Sachen Nutzerfreundlichkeit und insbesondere in der Optik aufwies. So vergingen die Wochen und der ursprüngliche Prototyp nahm an Kontur an.

Als im April 2008 die einmalige Gelegenheit bestand, einen Repräsentanten von Google™ mit dem inzwischen deutlich verbesserten Werkzeug zu beeindrucken, war das positive Feedback samt Verbesserungsvorschläge Grund genug, noch einmal alle Kräfte zu mobilisieren und den Prototypen »geolyzer« tatsächlich bis zum Juli auf die nächste Stufe zu bringen. Bereits zu dem Zeitpunkt war mir als Entwickler der Software allerdings klar, dass die Konzeption und Integration der Hauptkomponenten zeitlich gesehen unmöglich wird. Darum beschloss ich, diesen Schritt für die Diplomarbeit aufzusparen und die letzte Hürde allein zu bewältigen.

Mit dem Ende des Kurses trennten sich im Juli 2008 die Wege und ich möchte mich hiermit noch einmal ausdrücklich bei Frank Brünjes für die konzeptionelle Mitarbeit sowie bei Verena Batz für die gestalterische Leistung bedanken. Ohne eure Mitarbeit hätte der geolyzer noch immer einen unaussprechlichen Namen sowie eine fahle Optik.

Weiterhin bedanke ich mich bei Prof. Dr. Thorsten Teschke und Dr.-Ing. Volker Wittpahl für die hervorragende Betreuung während der Arbeit. Ein besonderer Dank gilt den nuterra Gründern Jasmin Grigutsch, Wiebke Wybrands und Nicolas Pauluhn für den Praxistest und der wertvollen Evaluation des geolyzer. Ein weiterer Dank geht an Thiemo Eddiks, dem Art Director der visuamundo GbR, für den finalen Usability-Test zur Bewertung der auto-evaluierenden Komponenten.

Mit dieser Diplomarbeit und dem letzten Schliff am geolyzer schließe ich mein Informatikstudium an der Hochschule Bremen ab. Die letzten vier Jahre haben mich wie keine andere Phase meines Lebens beeinflusst, in menschlicher und in fachlicher Hinsicht, und erst langsam realisiere ich den Wert, den ein Hochschulstudium mit sich bringt.

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Bremen, 13. Oktober 2008

Dirk Köster

“A software should be adapted to the needs of the user, not the other way round.”

David Banes
AbilityNet

1. Einführung

Dieses Werk widmet sich einem Problem, welches im betreffenden Millieu allgemein bekannt ist, bei dem man nach Lösungsansätzen bislang aber noch vergeblich sucht. Um es zu sichten bewegen wir uns durch die Landschaft der Informationstechnologie, streifen die Geoinformatik, ziehen an Altbekanntem vorbei und gelangen schließlich zu einer inzwischen nicht mehr ganz so neuen Technologie, die ortsbezogene Daten als für Nutzer leicht zu verstehende Visualisierung aufbereitet. Es geht um Geobrowser, auch als Erd-Betrachter betitelt, mit denen sich nicht nur unser blauer Planet aus der Ferne anschauen lässt, sondern in denen auch – wie bei gewöhnlichen Webbrowsern – von jedermann Inhalte für andere bereitgestellt werden. Erahnen Sie das Problem bereits?

Nein, um zu viel Reklame im Geobrowser geht es nicht. Es geht viel eher um die Untersuchung und Verbesserung der Gebrauchstauglichkeit. Gerade bei dieser doch recht jungen Informationstechnologie finden sich aufgrund neuer Strukturen ähnliche Usability-Probleme für Anwender wie beispielsweise zu den Anfängen des Internets. Maximen, Methoden und Werkzeuge für Webseiten gibt es zuhauf – allerdings noch kein einziges Utensil, welches speziell die Nutzbarkeit im Geobrowser untersucht und gegebenenfalls verbessert.

Kein einziges? Wäre nicht geolyzer als datenbankbasierter Ansatz speziell für Google Earth präsent, dann gäbe es keines. Und um diesen noch unvollständigen Prototypen geht es in dieser Diplomarbeit, denn er soll basierend auf einem Usability-Konzept für Geobrowser im Praxisteil so komplettiert werden, dass am Ende gilt: es gibt ein Usability-Werkzeug für Geobrowser!

1.1 Problemdarstellung

Zu den aktuellen Haupttrends des Internets gehört die Nutzung von **Geobrowsern**. Dabei handelt es sich um **virtuelle Globen** zur Visualisierung von Geodaten und ebenso zur Verknüpfung eigener Inhalte über geografischen Kontext. Unterschieden wird primär zwischen **2D-Browsern**, wie z.B. **Google Maps** oder **Map 24**, in denen sich durch vorhandenes Kartenmaterial horizontal und vertikal navigieren lässt, und **3D-Browsern**, die eine dreidimensionale Navigation auf einer virtuellen Erdkugel erlauben. Mit mehr als 200 Millionen Nutzern weltweit ist **Google Earth** der derzeit am meisten verbreitete 3D-Geobrowser [REGIS]. Trotz des aktuellen Booms ist das volle Potenzial bei weitem noch nicht ausgeschöpft und viele Anwendungsfälle noch gar nicht erschlossen. Institutionen, Unternehmen als auch Privatpersonen erkennen langsam das Innovationspotenzial, das z.B. die **Keyhole Markup Language (KML)** als Instrument zum Verknüpfen georeferenzierter Inhalte mit sich bringt – sei es nun die dynamisch aufbereitete Kriminalstatistik einer Großstadt, Geschäftsideen für Tourismus⁽¹⁾ und Marketing oder ein mithilfe von Ortsmarken verfasster Lebenslauf.

Eine der Schwächen bleibt allerdings die **mangelnde Usability**. Einerseits gelten Geobrowser an sich als schwer zu bedienen, andererseits fehlen passende Konzepte und Werkzeuge zur Aufbereitung und Überprüfung des Inhalts. Bei einer stetig wachsende Menge an veröffentlichtem Material fehlt es oft an einer gut durchdachten Ordnerstruktur, schlecht platzierte Ortsmarken vermengen sich mit 3D-Gebäuden, konfuse Navigationswege führen selten zum Ziel und nicht einheitliche Icons erschweren die Orientierung für Nutzer.

Usability und Geobrowser haben offensichtlich noch nicht zueinander gefunden – ähnlich den Anfängen des Internets, wo weder Methodik noch Werkzeug zum gebrauchstauglichen Entwurf von Webseiten vorhanden war. Zum aktuellen Zeitpunkt existiert für Geobrowser keine zugeschnittene Methodik und ebenso wenig ein geeignetes Werkzeug, um speziell die Nutzbarkeit der Inhalte zu testen und gegebenenfalls zu verbessern.

(1) Das Reisebüro ITWG (<http://www.itwg.com>) als ein Beispiel bereitet ihr Hotelangebot für Italien als KML-Inhalte auf. Siehe dazu: www.de.itwg.com/citymap5/de/placemarks.kml

1.2 Zielsetzung

Damit Geobrowser in der Bedienung mehr den Bedürfnissen der Nutzer entsprechen, sind die jeweiligen Hersteller gefordert, ihre Produkte künftig gebrauchstauglicher zu gestalten. Für die gebrauchstaugliche Aufbereitung der Inhalte müssen **Richtlinien** und **Werkzeuge** her, die sich speziell der Usability-Problematik im Geobrowser stellen. Hiermit ist der Ansatz dieser Arbeit genannt: es ist zu untersuchen, ob und wie bestehende Usability-Praktiken der Software- und Webentwicklung sich auf den Geobrowser-Bereich übertragen lassen. Welche Methoden könnten übernommen werden, welche eignen sich weniger und welche sind eventuell zu modifizieren. Und gibt es darüberhinaus Ansätze, die einen neuen Weg gehen und bislang in der Praxis noch nicht vorhanden sind?

Basierend auf dieser Untersuchung ist ein Werkzeug zu erschaffen, welches diese Praktiken umsetzt – und mit dem Prototypen **geolyzer** als Ergebnis einer Vorstudie des an der Hochschule Bremen gehaltenen Kurses »Webpräsenzen in Geobrowsern« ist eine Basis vorhanden, die sich zu diesem Zweck gut eignet. Dabei realisiert **geolyzer** als noch nicht in der Praxis getesteter Prototyp zum aktuellen Zeitpunkt nur das Fundament, auf dem das Konzept und die Realisierung zur Untersuchung und Verbesserung der Usability errichtet werden soll.

Als datenbankbasierte Windows-Anwendung speziell für den 3D-Geobrowser Google Earth soll **geolyzer** die aktuell vorhandene Usability-Lücke schließen und eine zum aktuellen Zeitpunkt einmalige, einfach zu bedienende und darüberhinaus auch kostengünstige Lösung bereitstellen, um objektiv die Nutzbarkeit von Inhalten in Google Earth zu untersuchen und zu bewerten. Der Prototyp **geolyzer** ist bereits jetzt in der Lage, Usability-Tests mit Probanden zu vorgegebenen Geobrowser-Inhalten durchzuführen und die Nutzeraktivität teilweise auch aufzuzeichnen. Als zu entwickelnde Usability-Komponenten stehen automatisch erstellte **Visualisierungen** und **Analysen** und **Bewertungen** des **Nutzerverhaltens** auf dem Plan, die genaue Auskunft über Navigationslücken geben, als Report exportiert werden und Entwicklern helfen, ihre Geobrowser-Präsenzen in Zukunft gebrauchstauglicher zu gestalten.

1.3 Vorgehen

Um die genannten Ziele zu erreichen, müssen die nötigen Grundlagen vermittelt werden, um einerseits zu verstehen, was es mit dem aktuellen Trend der Geobrowser auf sich hat und warum diese wohl zu »den« Web-Innovationen des 21. Jahrhunderts zählen – und aller Wahrscheinlichkeit nach einen enormen Einfluss auf Privates, Kulturelles und Berufliches ausüben werden. Daraufhin ist zu verdeutlichen, warum die Untersuchung der Usability überhaupt wichtig ist, welche Prinzipien gelten und welche Verfahren sich hier bewährt haben. Um diese beiden Themengebiete zu verknüpfen, muss auf die Frage eingegangen werden, welche potenziellen Navigationsschwierigkeiten beim Geobrowser existieren und warum zur Verbesserung der Inhalte ein anderer Ansatz vonnöten ist, als es beispielsweise bei Webseiten der Fall ist.

Um die Bestandsaufnahme abzuschließen, widmet sich ein Kapitel dem Ist-Zustand des **geolyzer**. Wie erwähnt, existiert die Software als Prototyp, der bis auf die wichtigen Kernkomponenten – damit sind die Usability-Funktionen gemeint – so gut wie fertig entwickelt ist. Es wird auf die bereits vorhandene Funktionalität und auf die Softwarearchitektur eingegangen. Auch werden die verwendeten Technologien vorgestellt, da mit diesen die Implementierung im Praxisteil vollzogen wird. Zuletzt wird der wichtige Soll-Zustand für das Endprodukt definiert.

Basierend auf den erarbeiteten Ergebnissen wird ein Konzept für die fehlenden Usability-Komponenten erstellt, um diese zu realisieren und in die bestehende Architektur des **geolyzer** zu integrieren. Neben den nötigen Softwaretests erfolgt eine Evaluation mit Probanden in Zusammenarbeit mit lokalen Unternehmen, um einerseits das Programm unter realen Bedingungen zu testen und weiterhin erste Nutzerdaten zu erhalten, so dass in den letzten beiden Kapiteln reflektiert wird, ob der Aufwand sich gelohnt hat und der **geolyzer** tatsächlich als neuer Ansatz zur Untersuchung und Verbesserung der Gebrauchstauglichkeit für Geobrowser-Inhalte taugt.

“Digital Earth was always intended to allow us to fly from space down through higher resolution data sets to hover above any point on the Earth’s surface — and then display information relevant to that location from an infinite number of sources“

Albert Arnold Gore, Jr.
45th Vice President of the United States

2. Georeferenzielles Web

Den Trend des »Mitmach-Webs 2.0« gibt es seit geraumer Zeit: Nutzer stellen Eigenes online, erweitern fremde Inhalte, bilden Online-Gemeinschaften und ...brandmarken ihre Mitmenschen im virtuellen Globus. Nun, auch der Bedarf an Intriganz wird im neuen Web offenkundig befriedigt, und es ist problemlos möglich, die eigenen Nachbarn mit einer Ortsmarke zu verlinken und anonym Positives, in der Realität aber meist Negatives, zu verfassen. Betrachtet man solche Dienste⁽²⁾, so stellen sich zwei Fragen: erstens, was kann man bei der Diagnose »Online-Steckbrief« tun? Zweitens, spricht dies für den Erfolg der Geobrowser?

Nicht unbedingt, aber die Vorhersage, dass Geobrowser zu »den« Web-Innovationen des 21. Jahrhunderts zählen und einen enormen Einfluss auf Arbeits- und Freizeitgewohnheiten der Bevölkerung haben, bestätigt sich zunehmend. Als im August 2005 die Berichterstattung zu »Hurricane Katrina« im Internet als auch im Fernsehen mithilfe von Google Earth und NASA World Wind erweitert, und der Verlauf sowie das Ausmaß der Zerstörung durch georeferenzielle Inhalte dynamisch dargestellt wurden, zeigte sich das Innovationspotential und der Nutzen der neuen Earth Viewer [SCHAR 2007]. Dabei stellen solche Szenarien – die Anreicherung von Nachrichten um Geo-Inhalte – nur einen Bruchteil der vielen Möglichkeiten, die letztendlich mit dieser Technologie machbar sind.

Um sich ein Bild über die momentane und künftige Bedeutung der Geobrowser zu machen, sind die Technologien vorzustellen und einige Grundbegriffe aus der Geoinformatik einzuführen. Abschließend wird auf die Problematik der Usability bezüglich Geobrowsern eingegangen – warum gelten Geobrowser als schwer zu bedienen, was unterscheidet die Inhalte von normalen Web-Inhalten, wie wird navigiert und wie wirkt sich all dies auf ein Konzept zur Verbesserung der Usability aus?

(2) www.rottenneighbor.com ist ein fragwürdiges Mashup aus Google-Maps und einer Community zum Denunzieren von Nachbarn. Es erfreut sich in den Vereinigten Staaten, inzwischen aber auch in Deutschland, großer Beliebtheit.

2.1 Trend zum Geobrowser

Ursprünglich waren Geobrowser eher für Akademiker und Verwaltungsbeamte interessant und beschränkten sich auf die rein informative Darstellung messbarer Werte. Diese als **Geoinformationssystem** (kurz *GIS*) bekannten Werkzeuge zur Verwaltung raumbezogener Daten stellten Informationen wie z.B. Umwelt- oder sozio-kulturelle Daten über zweidimensionalem Kartenmaterial dar und haben an vielen Instituten Analysten beim **Erfassen, Reorganisieren** und **Modellieren** von **Geodaten** unterstützt. Er wurde noch nicht daran gedacht, dass diese Technologie auch in der breiten Masse ihre Resonanz findet. Warum sollte der Normalbürger sich auch für ortsbezogene Metadaten wie Schädlingsbefall oder der Energie-Infrastruktur interessieren? Und so waren Visionäre gefordert, die das Innovationspotenzial erkannten und sehr wohl wussten, dass es für diese Technologie genug Anwendungsfälle gibt, die ein öffentliches Interesse wecken.

Durchbruch der neuen Technologie

So begann der Aufstieg 2005 mit Googles 2D-Geobrowser **Google Maps**, und aus einer Online-Alternative zu Stadtplan und Gelbe Seiten wurde im Laufe der Zeit weitaus mehr. Bereits 2004 kaufte Google das Programm **Keyhole** der gleichnamigen Keyhole Corp. auf und veröffentlichte es Mitte 2005 in der Basisversion als kostenfreien 3D-Geobrowser **Google Earth**⁽³⁾. Da Geobrowser nun durch das namhafte Unternehmen Google offiziell vermarktet die dritte Dimension betreten sollten, erkannten selbst Skeptiker, dass die Existenz solcher Werkzeuge im Web des 21. Jahrhunderts nicht wegzudenken wird. Mit flächendeckenden Satelliten-Bildern, Mess- und Suchfunktionen sowie der Markierung von Standorten wurde Nutzern eine reichhaltige Funktionalität in die Hand gegeben, mit der sich im Privaten und Beruflichen neue Möglichkeiten eröffnen.

Als Google auch das im CAD⁽⁴⁾-Bereich durch einfache Benutzung bekannt gewordene **SketchUp** aufkaufte und das Erstellen eigener 3D-Objekte für Google Earth zum Kinderspiel wurde, wuchs rasch eine Community, die für Google Earth genügend Nachschub an 3D-Inhalten produziert. Hiermit gelang es Google, die Vermarktung eigener Produkte an Anwender auszulagern.⁽⁵⁾

(3) <http://earth.google.de/>

(4) **CAD** – Computer Aided Design

(5) Dieser Vorgang wird auch als »Crowdsourcing« bezeichnet.

So ist es auch nicht verwunderlich, dass mit der schnellen Veröffentlichung der **Google Earth COM API**⁽⁶⁾ Programmierern eine freie Schnittstelle zur Verfügung gestellt wurde, um das Programm in eigene Applikationen zu integrieren.

Schattenseiten

Jedes Licht wirft Schatten, und so entstanden mit der freien Veröffentlichung von Satellitenbildern auch neue Probleme. Nicht nur der **Datenschutz**⁽⁷⁾ ist weiterhin ein heikles Thema, auch **Sicherheitslücken** wurden offengelegt. So musste im militärischen Bereich etliches Kartenmaterial zum Teil unkenntlich gemacht werden, z.B. die meisten Militäranlagen der USA – aber auch der NATO Luftwaffenstützpunkt Geilenkirchen⁽⁸⁾ als Heimat der AWACS-Aufklärungsflugzeuge ist für Terroristen nicht mehr per Geobrowser auszuspähen.

Geobrowser heute

Auch wenn Geobrowser stark im Kommen sind, so beschränkt sich die private Nutzung meist noch auf das Betrachten der eigenen Umgebung im virtuellen Globus. Das liegt einerseits daran, dass Geobrowser wie Google Earth nicht das volle Potenzial ausschöpfen und mit den über dem Globus verteilten Ortsmarken bislang nur aufklappbare Inhaltsfenster bereitstellen, in denen Metadaten zu einer geografischen Koordinate abgelegt sind. Andererseits ist auch vielen Anwendern noch nicht klar, was mit dieser Technologie machbar sein wird. Dennoch gibt es bereits jetzt Beispiele, wo mehr Potential ausgeschöpft wird.

- » **Placeopedia**⁽⁹⁾ verknüpft Wikipedia-Artikel mit Google Maps.
- » **Flickrvision**⁽¹⁰⁾ zeigt Fotos an der geografischen Position, wo sie auch gemacht wurden.
- » **REGIS:GE**⁽¹¹⁾ ist eine interaktive Anwendung zur Visualisierung historischer Geodaten.
- » Das **Wireless Communication Lab**⁽¹²⁾ in Baltimore stattet Busse mit GPS und WLAN aus, die ständig ihre Position und Geschwindigkeit an einen Server übertragen, was in Google Earth durch eine entsprechende KML für jedermann zu betrachten ist.

(6) **API** – Application Programming Interface

(7) siehe dazu <http://www.shortnews.de/start.cfm?id=710170>

(8) siehe dazu <http://www.betrachter.de/zensur-nato-luftwaffenstutzpunk-geilenkirchen/>

(9) www.placeopedia.com/

(10) <http://flickrvision.com/>

(11) http://benneten.de/referate/2007-07-26-regis-ge_tu/referat_regis-ge_tu_benneten.de.pdf

(12) siehe dazu: <http://wireless.cs.jhu.edu/>

Weitere Anwendungen und Geschäftsmöglichkeiten existieren als Konzept, sind angekündigt oder teilweise bereits realisiert [GE THEMA].

- » **Ortsbezogene Produkte und Dienstleistungen** wie Miet- und Kaufobjekte von Immobilienportalen werden in Google Earth visualisiert und per Hyperlink mit der entsprechenden Webseite verknüpft.
- » **Geschäftsinformationen mit Raumbezug**, seien es **RSS-Feeds**⁽¹³⁾ oder Reiserouten, werden in Google Earth platziert.
- » Im Sportbereich können Anlässe auch auf virtuellen Globen mitverfolgt werden.

Blick in die Zukunft

...Wird Web 3.0 dreidimensional? Nach Ansicht von John Hanke, dem Leiter der **Google Earth Division**, wird dies der Fall sein: *»Google Earth is turning into a map of historical significance. It is going to be a map of the world that is more detailed than any map that's ever been created [GEOWEB].«*

Mögliche **soziale** und **kulturelle Auswirkungen** werden ebenso genannt:

- » **Ortsbasierte Suche** wird eine Alternative zur herkömmlichen Suche.
- » Geobrowser bringen Menschen mit ähnlichen Interessen zusammen und formen neue **Gemeinschaften**.
- » **Ortsabhängige Dienste in mobilen Geobrowsern** verändern die Alltagsgewohnheiten der Menschen.
- » Geobrowser werden mit fortschreitender Verknüpfung georeferenzierter Inhalte eine **virtuelle Realität** erschaffen, die unsere Wirklichkeit weitestgehend abbildet.

Und **Google Earth** belegt momentan bei all dem noch den größten Marktanteil. Aber auch die Konkurrenz war nicht untätig und möchte mit **ArcGIS Explorer**⁽¹⁴⁾ und **NASA World Wind**⁽¹⁵⁾ am Erfolg der 3D-Geobrowser teilhaben. Ein Grund, die aktuell relevanten Technologien im nächsten Abschnitt näher zu beleuchten.

(13) **RSS** – Really Simple Syndication. Neue Inhalte werden dynamisch als XML kodiert und zugänglich gemacht.

(14) <http://www.esri.com/software/arcgis/explorer/index.html>

(15) <http://worldwind.arc.nasa.gov/>

2.2 Überblick Earth Viewer

Als **Earth Viewer** bezeichnet man computergestützte Programme sowie webbasierte Dienste zur globalen Visualisierung der Erdoberfläche mittels verschiedener Geodaten. Dabei wird primär unterschieden zwischen **Web Mapping Systemen**, d.h. im Webbrowser verfügbare Funktionalität, und **Client Systemen**, die die Installation einer Software verlangen. Zum Letztgenannten zählen beispielsweise **NASA Worldwind** sowie **Google Earth** oder das Freeware-Produkt **Marble**⁽¹⁶⁾. **Google Maps** und **Map 24** hingegen sind typische Vertreter für Web Mapping Systeme.

Bezogen auf den praktischen Teil dieser Arbeit sind nur Client Systeme relevant⁽¹⁷⁾. Um den Datenaustausch eines Client Systems zu illustrieren, zeigt **Abbildung I** die prinzipielle Kommunikation eines solchen Systems mit den Datenservern.

Der Client verbindet sich über das Internet mit den jeweiligen Applikationsservern, z.B. denen von Google, welche wiederum mit mehreren Geodaten-Servern in Verbindung stehen und so Zugriff auf **Raster-, Vektor- sowie Indexdaten** erhalten.

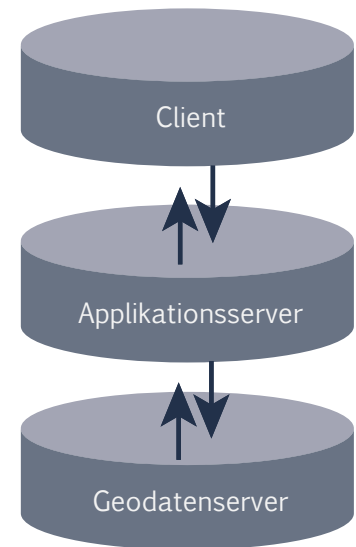


ABBILDUNG I:
Kommunikationswege



2.2.1 Google Earth

Das derzeit am meisten verbreitete Betrachtungsprogramm für Geodaten existiert in mehreren Versionen, wobei **Google Earth** die kostenlose Basisversion der Software ist. Zu den kostenpflichtigen Versionen zählt **Google Earth Plus** mit einem Tarif von 20 US-Dollar pro Jahr. Dafür gibt es auch ein Plus an Funktionalität:

- » GPS-Integration
- » höhere Auflösung beim Druck
- » Erstellung von Gebäuden

(16) Das Programm soll im Rahmen dieser Arbeit nur genannt werden. Siehe dazu <http://edu.kde.org/marble/>

(17) geolyzer ist ein reines Client-System, welches die Installation von Google Earth erfordert.

Für den professionellen Einsatz existiert **Google Earth Pro**. Zu einem Preis von 400 US-Dollar pro Jahr werden Zusatzmodule zum Erzeugen von Filmen sowie der Import von Datenformaten aus dem Bereich der professionellen Geoinformationssysteme angeboten.

Dreh- und Angelpunkt sind bei allen Versionen die Geodaten, die sich in **Bild-, Vektor- und Indexdaten** untergliedern lassen. Laut einer aktuellen Schätzung kann dabei Google Earth auf einen **Gesamtdatenbestand** von **150 TB**⁽¹⁸⁾ zugreifen. Besonders interessant ist die Tatsache, dass die Satelliten- und Luftbilder im Gegensatz zu anderen Earth Viewern aus vielen **unterschiedlichen Quellen** stammen. Aufgrund dieser Tatsache gestaltet es sich äußerst schwierig, die Bilder nahtlos in das Gesamtsystem zu integrieren. Und bei vielen Regionen, wo hochauflösende Bilder direkt an Aufnahmen geringerer Qualität grenzen, erkennt der Betrachter sofort den Qualitätssprung. Die folgende Tabelle gibt Auskunft über Herkunft und Auflösung des wichtigsten Bildmaterials [COMMONGIS].

Anbieter	Auflösung	Verfahren
MDA Federal	30m/Pixel	Satellit (Landsat TM)
Digital Globe	15m/Pixel	Satellit
Digital Globe	60cm/Pixel	Satellit (Quickbird 2)
Sanborn	30cm/Pixel	Flugzeug mit Luftbildkamera
Sanborn	15cm/Pixel	Flugzeug mit Luftbildkamera

TABELLE I: Herkunft und Auflösung des wichtigsten Google Earth Bildmaterial

Neben der Funktionalität zum Betrachten der Erdoberfläche gibt es die Möglichkeit, verschiedene Ebenen wie Staatsgrenzen oder Straßenverläufe ein- und auszublenden, eine Adress- und Branchensuche zu starten, Routenplanungen durchzuführen, Messfunktionen zu nutzen sowie eigene Ortsmarken zu erstellen und diese im KML-Dateiformat zu exportieren.

Seit Juni 2008 existiert zusätzlich zum Client System das **Browser Plugin**⁽¹⁹⁾, so dass Google Earth fortan ebenso im Webbrowser nutzbar ist. Um schnell eine große Menge an Nutzern zu erreichen, können eigens erstellte Google Maps-Inhalte auch über das Browser Plugin angezeigt werden. Derzeit wird noch nicht die volle Funktionalität des Client Systems angeboten, des Weiteren ist das Plugin noch auf **Windows-Systeme** beschränkt und zum aktuellen Zeitpunkt nicht kompatibel zu **Mac-Systemen**.

(18) siehe dazu http://www.earthblog.com/blog/archives/2006/09/news_roundup_google.html

(19) siehe dazu <http://code.google.com/apis/earth/>



2.2.2 NASA Worldwind

Bereits im Herbst 2004 erschien mit **NASA World Wind** ein direkter Konkurrent zu Google Earth. Auch wenn das Grundprinzip dasselbe ist, so bestehen doch einige Unterschiede. Einerseits ist der Umfang des Bildmaterials mit derzeit **5 TB** bei weitem nicht so hoch wie beim Mitbewerber, dafür stammt es aber auch aus **einer einzigen Quelle** und wirkt dementsprechend konsistenter. Nichtsdestotrotz fällt die geringe Auflösung des Standardsatelliten⁽²⁰⁾ **Landsat 7** von 15 Meter pro Pixel auf, und auch die Steuerung ist nach Ansicht vieler Nutzer etwas umständlicher als bei Google Earth [COMMONGIS].

Der Grund dafür liegt offensichtlich daran, dass eine andere Zielgruppe angesprochen wird. NASA World Wind fokussiert mehr die zugrunde liegenden Geodaten als die äußere Aufmachung des Programms. Und hier glänzt es: zusätzliches Kartenmaterial für Mond, Mars, Venus und Jupiter, Wetterradarbilder sowie Daten zur Sonnenaktivität sind besonders für Hobby-Astronomen und -Meteorologen interessant. Des Weiteren kann der Anwender zwischen verschiedenen Satelliten auswählen.



2.2.3 ESRI ArcGIS Explorer

Der letzte Erd-Betrachter im Bunde stammt von **ESRI** und ist von der Ausstattung her ähnlich zu den beiden bis jetzt genannten 3D-Geobrowsern: Satelliten- und Luftbilder, Vektordaten, Routenplanung, Messfunktionen, Ortsmarken, Darstellung anderer Himmelskörper sowie der Import von KML-Daten gehören zum Standard.

Der **Hauptunterschied** zu den beiden Konkurrenten ist die ausgeprägtere **wissenschaftliche Betonung**, die ArcGIS auch für Geologen interessant macht. **GIS-Funktionen** wie **Verschneidungen**, **Geokodierung** und Analyse von thematischen Daten zielen mehr auf den professionellen Anwenderbereich. Mit einem Programm wie Google Earth hingegen wird der nichtwissenschaftliche Anwender erreicht, da Inhalte hier viel umfangreicher visualisiert, interaktiv und multimedial vermittelt werden können.

(20) Nur in wenigen Großstädten der USA sind hochauflösende Satellitenbilder mit Auflösungen zwischen einem Meter und 25 Zentimeter pro Pixel verfügbar.

2.3 Georeferenzielles System

Der Begriff, der das beschreibt, was Geobrowser im Grunde leisten, lautet **Georeferenzierung**. Damit ist das Zuweisen raumbezogener Referenzinformationen zu einem Datensatz gemeint. Als Datensatz kommen alle Daten infrage, die in irgendeiner Form einen räumlichen Bezug haben: Umwelt- oder Verkehrsdaten, Infrastruktur, verfasste Artikel, Webseiten, Fotos, persönlicher und kultureller Kontext oder gar Spam⁽²¹⁾.

Bei der Art der Georeferenzierung unterscheidet man zwischen **Adresskodierung** – der Zuweisung einer Postanschrift zu einem Datensatz – und **Geokodierung** bzw. **Geotagging**. Beim Letztgenannten wird anstelle einer Adresse eine Koordinate dem Datensatz als Tag zugewiesen [WIKI GEOREFERENZIERUNG]. Die in Google Earth verwendete Ortsmarke (*engl. Placemark*), die allerlei Informationen, Bilder und Verlinkungen enthalten kann, ist ein Beispiel dafür, einen Datensatz mit einer Koordinate zu versehen.

Mit **Geotagging** ist derzeit die Idee verbunden, so viele Informationen wie nur möglich mit georeferenziellem Kontext zu verknüpfen⁽²²⁾ [SCHAR 2007]. Das Mammut-Projekt **WikiProjekt Georeferenzierung**⁽²³⁾ als ein Beispiel von mehreren arbeitet an der Anreicherung möglichst vieler Wikipedia-Artikel an räumlichen Daten. Bereits jetzt existiert in Google Earth unter »*Geografie im Web/Wikipedia*« eine eigene Ebene – und so ist es nicht unwahrscheinlich, dass künftig viele Wikipedia-Artikel alternativ über geografische Regionen anstelle von Suchbegriffen gefunden werden können.

Die Folgen, sollte irgendwann so gut wie jeder Punkt des Raumes mit Metadaten verlinkt sein, lassen sich nur schwer erahnen. So ist es nicht unwahrscheinlich, dass in etlichen Jahren **lokaler historischer** und **kultureller Kontext**, **persönliche Erlebnisse**, **gemeinschaftliche Ereignisse** und noch vieles mehr für jedermann mobil an jedem Ort der Erde abrufbar sein werden [SCHAR 2007]. Die Herausforderungen werden enorm, will man vermeiden, dass ungefilterter Inhalt Nutzer eher belästigt als nützt. Aber dies sind Fragen, die nicht Thema dieser Arbeit sind.

(21) Nach Ansicht von Experten könnte ortsabhängiger Spam durchaus zum Phänomen bei ortsabhängigen Diensten werden. [SCHAR 2007]

(22) Im Grunde sind alle menschlichen Artefakte in ihrer Historie stets mit einem Ort verknüpft. Dabei handelt es sich einmal um den Entstehungsort sowie die aktuelle Position.

(23) siehe dazu http://de.wikipedia.org/wiki/Wikipedia:WikiProjekt_Georeferenzierung

Die folgenden Unterkapitel beschreiben zwei gegenwärtig etablierte Methoden zur **Erzeugung georeferenzierter Inhalte** sowie das **georeferenzierende Modell**, auf welchem Geobrowser aufsetzen. Dieses ist einerseits zum Verständnis der Funktionsweise von Geobrowsern wichtig, andererseits um die **Keyhole Markup Language** besser zu verstehen.

2.3.1 Geoextraktion

Mindestens 20% aller Web-Ressourcen erlauben eine leicht zu erkennende und meist un-zweideutige geografische Identifikation [SCHAR 2007]. Nachrichten als einfachstes Beispiel haben stets einen Ursprungs- und einen Zielort – eine Nachricht der Frankfurter Allgemeinen über den Bundeswehreinsatz in Afghanistan hätte als Ursprungsort »Europa/Deutschland/Frankfurt« und als Zielort »Asien/Afghanistan/Kabul«.

Unter **Geoextraktion** (engl. *Geoparsing*) wird die Zuweisung von Geokoordinaten auf einzelne Worte oder Abschnitte in unstrukturierten Dokumenten verstanden. Das können z.B. Sätze wie »Fünf Kilometer nördlich von Berlin« sein, die mithilfe eines **Geoparsers** automatisch erkannt, übersetzt und in ein Geoinformationssystem eingepflegt werden. Problematisch ist es, wenn mehrdeutige Bezeichner das Ergebnis verfälschen – so mag »Bremen« eine Stadt in Norddeutschland, eine Kleinstadt in Ohio/USA oder der Nachname einer Person im Dokument sein.

2.3.2 Geokodierung

Im Gegensatz zur Geoextraktion, wo es um das Analysieren nicht zwingend eindeutiger Referenzen in unstrukturierten Dokumenten geht, bezieht sich der Prozess der **Geokodierung** (engl. *Geocoding*) allein auf das Analysieren eindeutiger Georeferenzen. Dazu zählen z.B. **Post-** oder **IP-Adressen** und **eindeutige Geokoordinaten** in Dokumenten oder Medien. Wurde z.B. ein Bild anhand der IP-Adresse eindeutig identifiziert, so kann es mit präzisen georeferenzierenden Koordinaten versehen und in ein Geoinformationssystem eingepflegt werden.

Der Webservice **www.Geocoder.com.ar** ist ein frei verfügbares Werkzeug, mit dem argentinische Geodaten wie Breiten- und Längengrad zu einer Adresse, einem öffentlichen Platz oder einer Sehenswürdigkeit via **XML**⁽²⁴⁾ abgefragt werden können.

(24) **XML** – eXtensible Markup Language

2.3.3 Georeferenzielles Modell

Der fachliche Ausdruck für die Weltkugel lautet **Geoid** – dabei handelt es sich um einen Ellipsoiden mit leichten Wellen, verursacht durch Unregelmäßigkeiten und Schwerefeld. Für Berechnungen und Visualisierungen wird meist ein vereinfachtes Modell eines **Referenzellipsoiden** genutzt. Das ist ein abgeflachtes Rotationsellipsoid, das als Annäherung an die ideale Erde dient. Google Earth als auch andere Geobrowser nutzen **WGS84** (World Geodetic System 1984) als global angepasstes **Rotationsellipsoid**. Die Bezeichnung bezieht sich einerseits auf das **geodätische Datum**⁽²⁵⁾ sowie auf den verwendeten Referenzellipsoiden [WIKI GE].

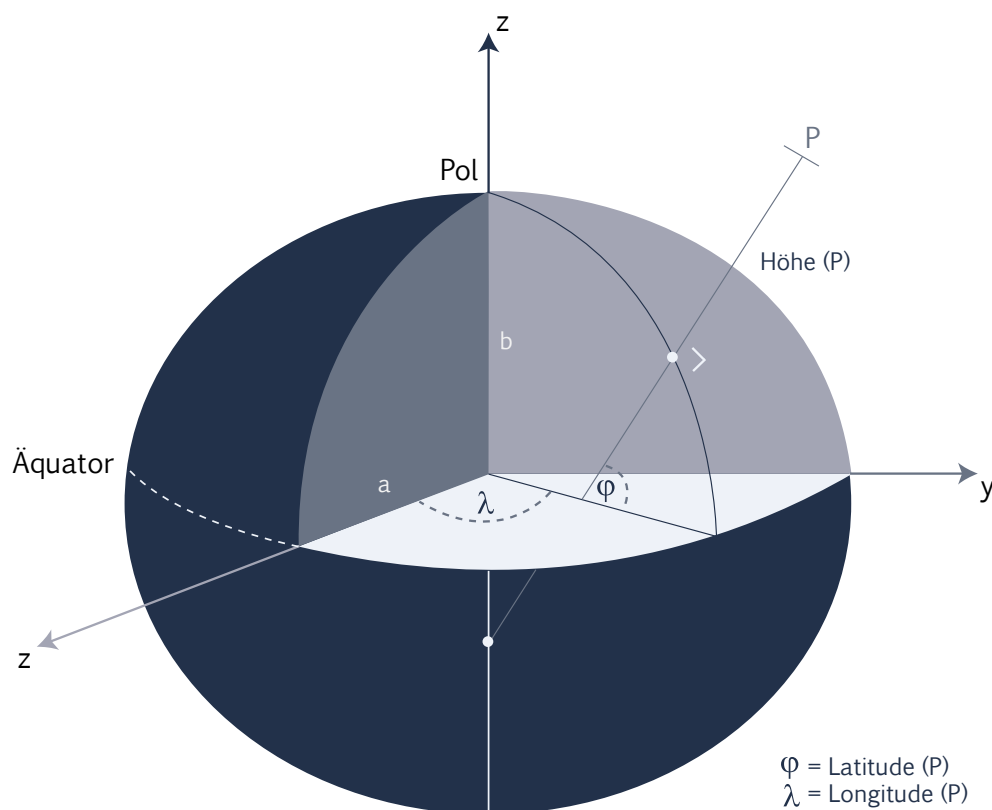


ABBILDUNG II: Rotationsellipsoid als vereinfachtes Modell der Erde [COMMONGIS]

(25) Das geodätische Datum ist eine Angabe, die die Abbildung von Positionen auf der unregelmäßig geformten Erdoberfläche auf ein Rotationsellipsoid parametrisiert.

Abbildung II zeigt das Modell dieses Referenzellipsoiden. Die Länge der großen Halbachse a beträgt 6378.137 km und die kleine Halbachse b hat eine Länge von 6356.752 km. Als **Nullmeridian** ist der Meridian von Greenwich definiert und der Äquator ist der **Hauptbreitenkreis**. Bei den Koordinaten, die einen Punkt P bestimmen, handelt es sich um **geographische Koordinaten**. Diese bestehen aus dem **Breitengrad** φ (*lat. latitudo, eng. latitude*) sowie dem **Längengrad** λ (*lat. longitudo, eng. longitude*). Der Breitengrad bestimmt im Winkelmaß die nördliche oder südliche Entfernung eines Punktes P zum Äquator. Der Wertebereich liegt somit zwischen 0° direkt am Äquator und $\pm 90^\circ$ an den Polen. Der Wertebereich des Längengrads beginnt am Nullmeridian mit 0° und geht in westlicher Richtung bis -180° sowie in östlicher Richtung bis 180° .

Etwas verwirrend für nicht eingeweihte Nutzer ist, dass die Darstellung der Geo-Koordinaten aus historischen Gründen prinzipiell durch mehrere Formate möglich ist. Google Earth unterstützt die gängigsten Formate:

- » **Dezimalformat** (51.524511° , 6.925238°)
- » **DMS-Format** mit Grad, Minuten und Sekunden ($51^\circ 31' 28.24''$, $6^\circ 55' 30.86''$)
- » **Grad, Dezimalminuten** ($51^\circ 31,471' \text{ N}$, $6^\circ 55.514' \text{ O}$)
- » **Transversales Mercator-Koordinatensystem** (Zone 32 N 5710195 m E 356069 m)

Als zusätzliche Komponente eines Punktes kommt die **Höhe** hinzu. Diese entspricht der Länge des senkrechten Vektors zwischen Erdoberfläche und Punkt P . Der Winkel zwischen Äquatorebene und dem Lot ist dabei der erläuterte Breitengrad φ . Da es sich beim Referenzellipsoiden nicht um eine Kugel handelt, verläuft das Lot, wenn man den Vektor verlängert, nur am Äquator und an den Polen durch den Erdmittelpunkt. Die tatsächliche Richtung zum Erdmittelpunkt wird als **geozentrische Breite** bezeichnet und weicht von der **geografischen Breite** um bis zu 0.2° ab.

2.3.4 Keyhole Markup Language

In der Geoinformatik werden seit langem unterschiedliche Datenformate für den Austausch von Geodaten benötigt. Da XML-basierte Verfahren des Datenaustausches und der Kommunikation inzwischen eine große Bedeutung haben und XML sich weltweit als Standard für die Interoperabilität von Anwendungen etabliert hat, war es naheliegend, dass für die Geoinformatik und Kartographie eine auf XML basierte Sprache namens **GML – Geography Markup Language** – entstanden ist. Mit dieser vom **OGC**⁽²⁶⁾ definierten Sprache wurde ein mächtiges Werkzeug geschaffen, um geografische Inhalte zu kodieren.

Aufgrund der Komplexität ist diese Sprache eher für wissenschaftliche Geoinformationssysteme geeignet und weniger dazu, eigene Inhalte als Ortsmarke zu veröffentlichen. Dafür existiert die **Keyhole Markup Language**, die nicht der **Kodierung** sondern der **Visualisierung** geografischer Inhalte dient. Mit der GML lässt sich beispielsweise ein Flußlauf mit allen relevanten Geoeigenschaften kodieren – die KML ist dazu nur eingeschränkt in der Lage, kann aber den Flußlauf über Vektordaten darstellen und mit einer Ortsmarke versehen, die Bilder und eine textuelle Beschreibung enthält.

KML basiert ebenso auf XML 1.0 und hat sich derzeit zu einem de facto Standard im Geobrowserbereich entwickelt, da auch die beiden Konkurrenzprodukte NASA World Wind und ArcGIS sowie Google Maps – allerdings mit Einschränkungen – die KML unterstützen. Auch wenn die Mächtigkeit im Vergleich zur GML geringer ausfällt, so es ist mithilfe der KML dennoch möglich, alle relevanten Elemente im Geobrowser zu kodieren: Vektordaten⁽²⁷⁾, Bilder, 3D-Modelle, Text-Informationen sowie die Kameraposition.

Um Medien wie Bilder, Icons oder durch Google SketchUp erstellte 3D-Objekte zusammen mit einer KML-Datei in einem Paket anderen Nutzern zur Verfügung zu stellen, existiert das **KMZ-Dateiformat**, welches nichts anderes als eine mit dem **ZIP-Algorithmus** komprimierte Datei ist. Dies ist besonders dann sinnvoll, wenn Medien nicht über eine URL angesprochen werden können.

(26) Das Open Geospatial Consortium (OGC) ist ein internationales, gemeinnütziges Industriekonsortium mit dem Ziel, offene Standards in Form von Schnittstellen und Protokollen für einheitliche Zugriffsmethoden auf raumbezogene Informationen zu definieren.

(27) Derzeit werden Punkte, Linien und Flächen unterstützt. Bögen oder Ellipsen sind noch nicht im Angebot.

Nur ansatzweise im Funktionsumfang ist die Geokodierung. So ist es derzeit für die USA und Kanada möglich, anstelle von Geokoordinaten eine eindeutige Adresse anzugeben, welche umgewandelt wird. Die Genauigkeit hängt allerdings stark von der Beschaffenheit der jeweiligen Region ab. Ist diese z.B. stark bebaut, so ist die Wahrscheinlichkeit eher gering, durch Angabe von Straße und Hausnummer das richtige Haus zu erwischen [COMMONGIS].

Grundlagen

Aufgrund der XML-Abstammung beginnt eine KML-Datei mit dem bekannten XML-Header. Innerhalb des **KML-Stammelements** befindet sich der Bezug zum **Google Earth Namensraum** sowie alle Elemente des Dokuments.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.2">
  <!-- KML Elemente -->
</kml>
```

Wichtigstes KML-Element ist die **Ortsmarke**, die als Container für allerart Inhalt dient. Optional sind Name, Beschreibung und Kameraposition, obligatorisch ist die Angabe einer Koordinate innerhalb des **Point**-Tags, um die Ortsmarke auf eine Position auf dem Globus abzubilden.

```
<Placemark>
  <name>Bremen</name>
  <description>Kleinsten Stadtstaat Norddeutschlands</description>
  <Point>
    <coordinates>53.3262, 8.8151, 0</coordinates>
  </Point>
  <LookAt id="ID">
    <longitude>123.122</longitude>
    <latitude>75.2331</latitude>
    <altitude>0</altitude>
    <range>23.123</range>
    <tilt>0</tilt>
    <heading>0</heading>
    <altitudeMode>clampToGround</altitudeMode>
  </LookAt>
</Placemark>
```

Das `description`-Tag unterstützt eine Teilmenge der **HTML⁽²⁸⁾-Formatierungselemente**. Dazu wird dringend empfohlen, den HTML-Inhalt innerhalb des `description`-Tag mit `<![CDATA[...]]>(29)` einzukapseln.

Im Gegensatz zum obligatorischen `Point`-Tag definiert das optionale `LookAt`-Tag die Orientierung der Kamera, die auf die angegebenen Koordinaten der Ortsmarke gerichtet ist. `Longitude`, `Latitude` und `Altitude` bestimmen die Position, `range` den Abstand zu dieser Position, `tilt` den Winkel zwischen der Richtung der `LookAt`-Position und der Normalen zur Erdoberfläche. **Abbildung III** verdeutlicht dies. Das `heading`-Tag bestimmt die Kompassrichtung zwischen 0 und $\pm 180^\circ$ und mit `altitudeMode` wird festgelegt, wie `altitude` ausgewertet wird. Dabei stehen vordefinierte Werte einer Auflistung zur Wahl:

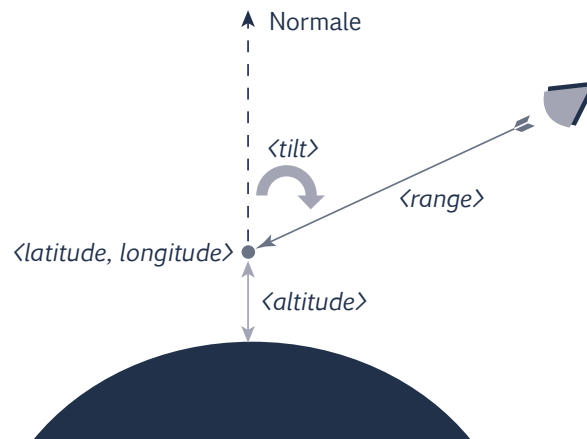


ABBILDUNG III:
Kameraposition und Manipulation in KML [KML]

- » **clampToGround** – Platziert die `LookAt`-Position auf dem Boden (*Standardwert*).
- » **relativeToGround** – Wertet `altitude` in Metern über dem Boden aus.
- » **absolute** – Wertet `altitude` in Metern über dem Meeresspiegel aus.

Darstellung und Gruppierung

Zur **Formatierung** von Ortsmarken steht ein reichhaltiges Angebot an Optionen bereit. Formatieren lässt sich das Info-Fenster über `balloonStyle`, der Name des Features mit dem `labelStyle`-Tag sowie das Icon über `iconStyle`. Diese Style-Elemente sind alle vom abstrakten Element `ColorStyle` abgeleitet.

(28) **HTML** – Hypertext Markup Language

(29) **CDATA** – Character Data. Der Inhalt wird nicht als Markup interpretiert.

Bei der Farbdefinition ist zu beachten, dass der in KML verwendete Farbraum das hexadezimale AABBGRRR-Format verwendet. Durch den Alpha-Wert (AA) kann somit die Deckkraft in Schritten von 0 (00) bis 255 (FF) angegeben werden, um bei Bedarf Transparenz zu erzeugen.

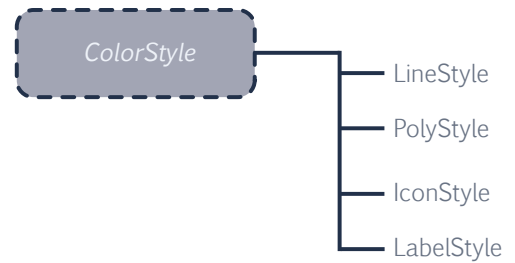
```
<Style>
  <LabelStyle>
    <color>ff0000cc</color>
    <colorMode>random</colorMode>
    <scale>1.5</scale>
  </LabelStyle>
  <IconStyle>
    <color>ff00ff00</color>
    <colorMode>normal</colorMode>
    <scale>1.1</scale>
    <Icon>
      <href>http://www.bilder.de/test.png</href>
    </Icon>
  </IconStyle>
  <BalloonStyle>
    <bgColor>FF0000FF</bgColor>
    <textColor>FFFFFFAA</textColor>
    <text>Freier Text</text>
  </BalloonStyle>
</Style>
```

Zur **Wiederverwendung von Stilen** und insbesondere zur Reduktion der Dateigröße ergibt es Sinn, Stile in das zum Einkapseln von Schemata verfügbare Document-Tag auszulagern, eine ID zu definieren und diese aus den einzelnen Placemarks über das `styleurl`-Tag zu referenzieren.

```

<Document>
  <Style id="meinStil">
    <LabelStyle>
      <color>FF0000FF</color>
    </LabelStyle>
  </Style>
  <Placemark>
    <styleUrl>#meinStil</styleUrl>
    <Point>
      <coordinates>23.5942, 36.64551, 0</coordinates>
    </Point>
  </Placemark>
</Document>

```



Um einzelne Ortsmarken zu einer logisch aufgebauten Hierarchie zu **gruppieren**, steht das `Folder`-Tag bereit, mit dem sich Verzeichnisse innerhalb des Browsers erzeugen lassen.

```

<Folder>
  <name>Norddeutschland</name>
  <Placemark>
    <name>Bremen</name>
    <Point>
      <coordinates>53.3262, 8.8151, 0</coordinates>
    </Point>
  </Placemark>
  <Placemark>
    <name>Hamburg</name>
    <Point>
      <coordinates>53.6262, 10.0051, 0</coordinates>
    </Point>
  </Placemark>
</Folder>

```

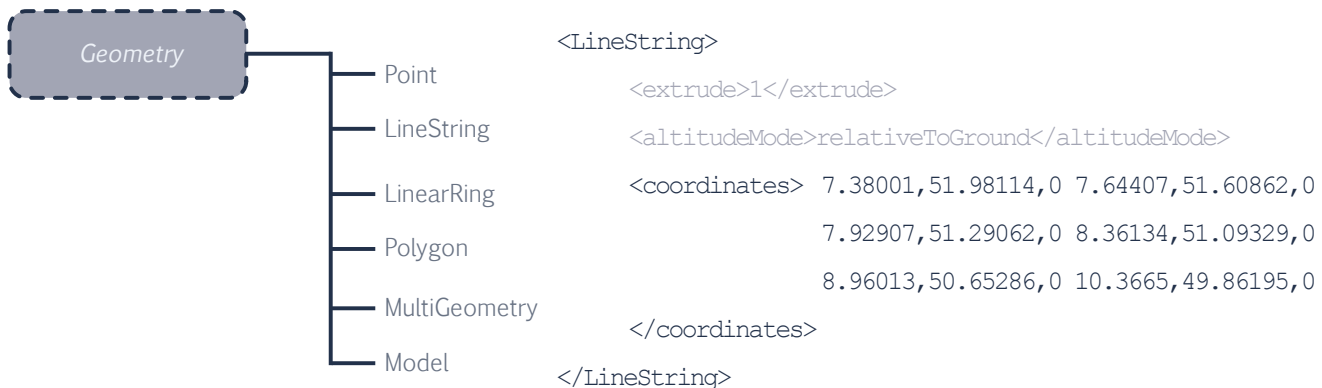


Geometrie-Objekte

Vom abstrakten Element `Geometry` sind einige **geometrische Primitive** abgeleitet, die zur Darstellung von Vektordaten im Browser geeignet sind. Das bereits erwähnte `Point`-Element bestimmt einen Punkt im Raum mithilfe von Breitengrad, Längengrad und Höhe. Durch das boolesche `extrude`-Tag lassen sich alle Geometrie-Objekte, die sich über dem Boden befinden, in die nächste Dimension extrudieren. Aus einem Punkt würde eine Linie, aus einer Linie eine Fläche und aus einer Fläche ein Volumenkörper. Voraussetzung ist, dass `altitudeMode` nicht mit `clampToGround` gesetzt ist.

```
<Point id="Point HH">
  <extrude>1</extrude>
  <altitudeMode>absolute</altitudeMode>
  <coordinates>53.6262, 10.0051, 12.24</coordinates>
</Point>
```

Um ein **Liniensegment** beliebiger Länge zu erstellen, sind 2 bis n Koordinatentupel vonnöten, die im `coordinates`-Tag innerhalb des `LineString`-Elements definiert und mit einem Leerzeichen voneinander separiert werden.



Soll das **Liniensegment geschlossen** sein, so ist auch dies mit dem `LinearRing`-Element möglich. Zu beachten ist hierbei, dass mindestens vier Koordinaten angegeben werden und Koordinate n mit Koordinate 1 identisch ist.

```

<LinearRing>
  <coordinates>
    -122.365662,37.826988,0 -122.365202,37.826302,0
    -122.364581,37.826554,0 -122.365038,37.827237,0
    -122.365662,37.826988,0
  </coordinates>
</LinearRing>

```

Ein **Polygon** wird durch eine äußere und optional durch eine innere Begrenzung definiert. Die Begrenzungen werden dabei durch das `LinearRing`-Element erstellt.

```

<Polygon>
  <outerBoundaryIs>
    <LinearRing> ... </LinearRing>
  </outerBoundaryIs>
  <innerBoundaryIs>
    <LinearRing> ... </LinearRing>
  </innerBoundaryIs>
</Polygon>

```

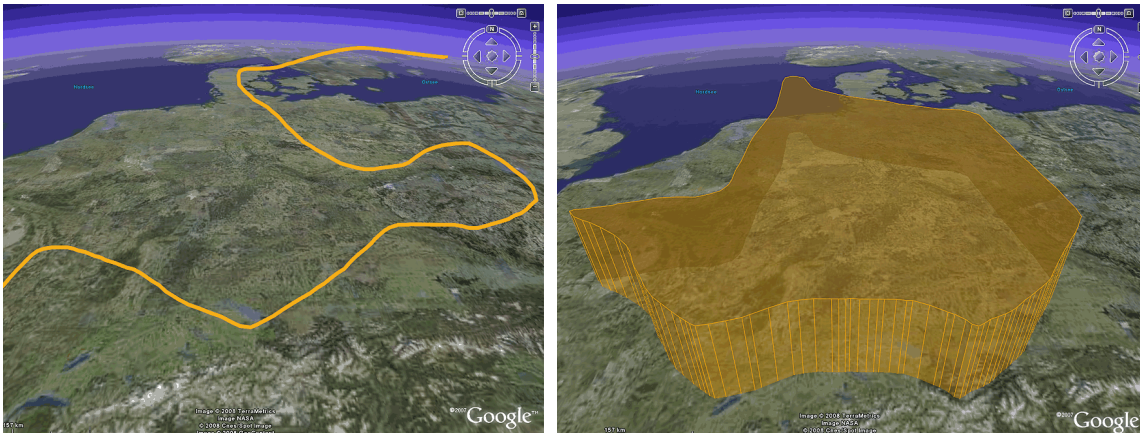


ABBILDUNG IV:

KML Geometrieobjekte LineString (links) sowie Polygon (rechts)
Satellitenbilder ©NASA, TerraMetrics, Cnes/Spot Image, GeoContent

Des Weiteren existiert zum **Gruppieren** von Geometrie-Objekten das `MultiGeometry`-Element, mit dem sich beispielsweise ein Gebäude als Komposition mehrerer Primitive erzeugen lässt. Das `Model`-Element ist für diese Arbeit wenig relevant und bezieht sich auf 3D-Objekte, die über ein `Link`-Tag referenziert werden.

Dynamisches Nachladen

Die Mächtigkeit der KML zeigt sich beim **dynamischen Nachladen** von **Inhalten**. Ein `NetworkLink` referenziert eine externe KML oder KMZ im Netzwerk. Diese kann durch den `refreshMode` in Kombination mit dem `refreshInterval`-Tag periodisch nachgeladen werden. Beispielhaft wurden dazu die Busse aus Baltimore genannt, die ihre aktuelle Position und Geschwindigkeit via WLAN an einen Server senden, der periodisch die Daten als KML online stellt. Über den `NetworkLink` kann eine KML-Datei auf diese Daten zurückgreifen und sie für den Anwender visuell aufbereitet im Geobrowser anzeigen.

```
<NetworkLink>
  <name>Update</name>
  <Link>
    <href>http://www.kmls.com/test/kml/test.kml</href>
    <refreshMode>onInterval</refreshMode>
    <refreshInterval>3</refreshInterval>
  </Link>
</NetworkLink>
```

So ist es mit der KML möglich, dass Anwender sich **dynamische Inhalte** ihres Geobrowsers nach eigenen Interessen zusammenstellen können. Interessiert sich der Nutzer nur für Sportereignisse, so könnte er seinen Geobrowser mit entsprechenden KML-Dateien so konfigurieren, dass nur aktueller Inhalt aus dem weltweiten Sportbereich angezeigt wird.

Damit sind die wichtigsten Eigenschaften der KML genannt. Für weiterführende Informationen steht eine übersichtlich verfasste KML-Referenz auf den Google Servern bereit. Diese Referenz diente ebenso als Quelle für dieses Unterkapitel [KML].

2.4 Geobrowser und Usability

Es wurde darauf hingewiesen, dass bei Geobrowsern eines der Hauptprobleme die mangelnde Usability ist. Beim Vergleich der aktuellen Earth Viewer wurde berichtet, dass **NASA World Wind** bezüglich der Steuerung von Anwendern überwiegend schlechte Wertungen erhielt. Und auch die neueste Version von **Google Earth 4.3**, die im April 2008 veröffentlicht wurde, hat auf den ersten Blick Anwender eher verschreckt als begeistert: die neue Steuerung ist ungewohnt und die Nutzung generell nicht mehr so intuitiv.

Mit der defizitären Steuerung allein ist das ganze Problem noch nicht ausreichend skizziert, denn das Zurechtfinden der Nutzer ist ebenso prekär. Bei einem Nutzertest⁽³⁰⁾ zur aktuellen Version von Google Earth wurden Probanden Aufgaben gestellt wie *»Suche den zweithöchsten Gipfel der Alpen«* oder *»Navigiere auf Grundniveau vom Münchener Hauptbahnhof zum Münchener Rathaus«*. Die Ergebnisse waren ernüchternd:

- » die **Informationssuche** offenbart sich ohne Hilfsmittel als so gut wie unmöglich
- » die **Navigation** an dem Nutzer noch unbekannten Orten ist schwierig
- » die **Gliederung** der Ebenen ist undurchsichtig
- » die **Suche in/nach Ebenen** ist ein Problem

Mit den genannten Erkenntnissen zur mangelhaften Gebrauchstauglichkeit im Geobrowser wird sich auf das **Werkzeug an sich** bezogen und nicht auf die **Inhalte**, die damit einhergehen – für diese sind die Hersteller nicht verantwortlich.

Gebrauchstaugliche Inhalte

Es ist bei Webbrowsern nicht anders: für Inhalte sind allein die Anwender verantwortlich, also Privatleute, Institutionen, Firmen etc. Und damit ist das **zweite Usability-Problem** genannt. Denn bei einer Flut an Produzenten für Inhalte bleibt die Gebrauchstauglichkeit meist immer auf der Strecke, und Methoden und Werkzeuge müssen her, die irgendwie Abhilfe schaffen. So ist es im Web-Bereich inzwischen geschehen, nur bei Geobrowsern hat sich noch nicht viel getan.

(30) Die genannten Erkenntnisse beruhen auf einem Usability-Test zu Google Earth, der mit ca. 20 Probanden am 21.04.08 im Rahmen des Kurses »Applikationsentwicklung in Geobrowsern« an der Hochschule Bremen durchgeführt wurde.

Während Inhalte bei Webbrowsern meist aus einer vertikal scrollbaren Ansammlung von Texten, Bildern und Verlinkungen bestehen, können solche im Geobrowser über den ganzen Globus verteilt und insbesondere **übereinander gelagert** sein. Zu den aufklappbaren Ortsmarken, die analog zu Webinhalten aus Texten, Bildern und Verlinkungen bestehen können, kommen im Geobrowser weitere Inhalte wie Icons, 3D-Objekte, Vektordaten sowie Ebenen mit zusätzlichen georeferenziellen Informationen hinzu.

Dabei werden gerade hier immer mehr Inhalte produziert und veröffentlicht. Firmen und Institutionen erkennen bereits jetzt, dass zusätzlich zum Web-Auftritt in einigen Jahren eine Geobrowser-Präsenz unabdingbar ist⁽³¹⁾, will man den Anschluss bewahren. Ob nun ein Touristik-Unternehmen sein aktuelles Angebot auf dem Globus visualisiert, ein Energieunternehmen für jede Region den aktuellen Strom-Tarif offenbart oder die Stadtverwaltung Auskunft über aktuelle Baustellen gibt – Inhalte werden für Anwender erstellt, und diese sollten so aufbereitet sein, dass sie zum Gebrauch auch taugen.



ABBILDUNG V:

KML-Inhalte negativ (links) und positiv (rechts) aufbereitet

Linke Abbildung ©U.S. House Defense 2008, rechte Abbildung ©nuterra

Satellitenbilder ©NASA, GeoContent

(31) Die TUI bereitet derzeit ihre Geobrowser-Präsenz vor, von Wetter.com stammt die Erweiterung **Wetter-plugin.kml** und das Züricher Tiefbauamt bietet eine dynamische KML mit aktuellen Baustellen an.

Die Realität sieht noch anders aus, betrachtet man öffentlich verfügbare KML-Inhalte. Meist ist keine Navigationsstruktur erkennbar und alle Ortsmarken sind unstrukturiert in der Datei lose aneinander gereiht – ohne Verzeichnisse und eher selten mit internen Verlinkungen. Nutzer wissen nicht, bei welchen Ortsmarken zu beginnen ist, welche von Relevanz sind und ob die Inhalte überhaupt auf eine Region beschränkt sind oder sich über den gesamten Globus verteilt erstrecken – ohne in der Google Earth Ordnerstruktur alle Ortsmarken einzeln durchzuklicken. Passendes Negativbeispiel hierzu ist die KML der US-Regierung⁽³²⁾, die die jährlichen Verteidigungsausgaben als Ortsmarken innerhalb der US-Bundesstaaten aufbereitet. Hier wird der Nutzer im aktuellen Jahr 2008 mit 1210 losen Ortsmarken konfrontiert, die den gesamten nordamerikanischen Kontinent auffüllen und bei Betrachtern für mehr Verwirrung als Durchblick sorgen.

Neben der Navigation wird ebenso die **textuelle** sowie die **optische Aufbereitung** der Inhalte vernachlässigt: Texte innerhalb der aufklappbaren Ortsmarken sind selten **lesefreundlich formatiert**, 3D-Objekte **überdecken** andere **Inhalte**, Bild-Overlays erscheinen schon aus großer Entfernung als unproportionierte Störfaktoren und die Möglichkeit, durch das `lookAt`-Tag allen Ortsmarken eine **passende Draufsicht** zu gewähren, wird meist ignoriert.

Dass es auch anders geht, zeigt das Bremer Universitäts-Spin-off **nuterra**⁽³³⁾, welches in Zusammenarbeit mit dem Oldenburger Gestaltungspartner visuamundo GbR⁽³⁴⁾ Inhalte für Kunden aufbereitet und individuelle Inszenierungen in Geobrowsern ermöglicht. Hier wird der Gebrauchstauglichkeit von Inhalten große Aufmerksamkeit gewidmet.

Webbrowser vs. Geobrowser

Um Inhalte für Geobrowser nutzerfreundlich zu gestalten, stellt sich die Frage, was bei der Navigation im Geobrowser anders als beim Webbrowser ist. Anders gefragt, wie navigiert ein Nutzer mit den Mitteln des Browsers und innerhalb des bereitgestellten Inhalts. Die folgende Tabelle stellt die gängigsten Navigationswerkzeuge beider Browserarten gegenüber.

(32) siehe dazu www.sunlightlabs.com/earmarks/house_defense_earmarks_08.kml

(33) <http://www.nuterra-bremen.de/>

(34) <http://www.visuamundo.com/>

Webbrowser

Navigation im Browser	Navigation im Inhalt
URL	Links
Zurück-Button	Navigationsmenü
Tabulatoren	Suche
Bookmarks	Sitemap
Suche	

TABELLE II: Navigationsmöglichkeiten im Webbrowser**Geobrowser**

Navigation im Browser	Navigation im Inhalt
Kompass zur Manipulation der Kamera	Links in Ortsmarken
Mausfunktionen zur Änderung der Ansicht	Ortsmarken Icons auf dem Globus
Listenansicht für Orte	Bild-Overlays
Ebenen-Filter	Vektor-Objekte
Suche	

TABELLE III: Navigationsmöglichkeiten im Geobrowser

Dem Nutzer, der mit Webbrowsern einigermaßen vertraut ist und zum ersten Mal Google Earth startet, wird auffallen, dass gewisse bekannte Steuerelemente einfach nicht vorhanden sind. Navigiert er durch Ortsmarken und will beispielsweise zur letzten Marke zurück, so sucht er den **Zurück-Button** im Geobrowser ohne Erfolg.

Auch die Ortsmarke an sich ist ein völlig neues Steuerelement, welches es dem Nutzer erlaubt, durch Inhalte zu navigieren. Während im Webbrowser dies durch eine überschaubare Anzahl an **Links** und **Menüs** geschieht, kann das Auftreten nicht strukturierter und besonders aus der Ferne chaotisch wirkender Ortsmarken Nutzer mehr verwirren als helfen. Sollten in Zukunft Ortsmarken voll HTML-tauglich werden, so könnten sie von einer gängigen Webseite nicht mehr unterschieden werden – und der räumliche Bezug bleibt nicht mehr abstrakt, sondern ist im Geobrowser für Nutzer greifbar.

Wer Inhalte bereitstellt, der sollte sich der Unterschiede bewusst sein und bereits im Vorfeld der Erstellung einer Webpräsenz im Geobrowser sich um die Lösung folgender Probleme Gedanken machen:

- » wie ist eine **Nutzerführung** am besten zu realisieren?
- » wie ist **Text** optimal **einzusetzen**?
- » wie nutzt man **Filter**, damit Inhalte erst bei Annäherung erscheinen?
- » wie lassen sich **Inhalte** geschickt **nachladen**?
- » wie sind **Inhalte** so **zu platzieren**, dass sie keine anderen Inhalte überdecken?

Ein guter Ansatz für eine gelungene Nutzerführung ist es, alternativ zur Navigation innerhalb der Google Earth Ordnerstruktur in der linken Navigationsleiste für eine überschaubare Navigation innerhalb der Ortsmarken zu sorgen – was sich z.B. durch Links realisieren lässt. Die KML stellt genug Formatierungsoptionen bereit, um Listen, Links und Menüs zu erzeugen, mit denen die Navigation innerhalb der Präsenz dem Nutzer transparent gemacht wird. Ebenso ist eine konsistente Verwendung von Icons für Nutzer hilfreich.

Fazit

Es wird konstatiert: Geobrowser leiden **allgemein** an **schlechter Nutzbarkeit**, und hier sind eindeutig die Hersteller gefordert, um ihre Produkte durch die Erkenntnisse der Software-Ergonomie zu verbessern. Zweites Problem: durch die wachsende Bereitstellung von Inhalten in Geobrowsern bleibt die **Gebrauchstauglichkeit der Inhalte** meist auf der Strecke – was vor allem daran liegt, dass im Geobrowser andere Navigationsstrukturen als im Webbrowser herrschen und dieses einen anderen Entwicklungsprozess erfordert. Die **Lösung** lautet **Usability**, d.h. es müssen **Richtlinien** zum **gebrauchstauglichen Entwurf** von **Inhalten** im Geobrowser her, und auch **geeignete Werkzeuge**, die speziell an die Gegebenheiten des Geobrowsers angepasst sind.

“The main goal is not to complicate the already difficult life of the consumer.”

Raymond Loewy
Father of Industrial Design

3. Usability

Wenn beim Software-Entwurf der Begriff »Usability« fällt, so verweisen Programmierer gern an den Designer, der dafür Sorge zu tragen hat, dass das zu entwickelnde Produkt vom Kunden irgendwie angenommen wird. Im guten Fall setzen sich Gestaltung, Marketing und Entwicklung zusammen, um gemeinsam die Firmenidentität auf einen klaren und gebrauchstauglichen Entwurf abzubilden, der bereits während der Entwicklung entsprechend getestet wird. Im schlechten Fall allerdings setzt der Designer seine eigene Vorstellung eines sich von der Masse abhebenden Konzepts durch und verkauft dieses den Vorgesetzten und Kollegen als »den« exklusiven Entwurf. Und spätestens bei der ersten Kundenabnahme wird klar, dass das Produkt zwar schick aussieht, aber völlig konfus in der Bedienung nicht eingeweihte Nutzer völlig überfordert.

Um solche Szenarien in Zukunft zu mindern, gibt es die Disziplin der Usability, die sich damit befasst, wie Produkte bereits während der Entwicklung ein höheres Maß an Gebrauchstauglichkeit erfahren. Um sich von der Materie ein entsprechendes Bild zu machen, wird zuerst auf die Bedeutung der Usability für allgemeine Produkte eingegangen und mit dem gängigen Vorurteil aufgeräumt, Usability sei ein geistiges Konstrukt des neuen Jahrtausends – und ist nachträglich durch Abhaken einiger Regeln jedem Produkt mühelos aufzutragen. Danach spezialisiert sich die Betrachtung immer mehr von Produkten zu Inhalten und es werden auf dem Markt befindliche Werkzeuge vorgestellt, um abschließend zu reflektieren, inwieweit die vorgestellten Methoden und Maximen sich als Usability-Ansatz für Geobrowser eignen.

3.1 Gebrauchstauglichkeit von Produkten

Mit dem aus den Worten »Use« (*benutzen, Gebrauch*) und »Utility« (*Gebrauchs..., Nützlichkeit*) zusammengesetzten Begriff **Usability** verbinden die meisten Entwickler ein einfach zu erlernendes Produkt⁽³⁵⁾, welches dem Nutzer verständlich ist und in der Bedienung keine Schwierigkeiten bereitet. Wenngleich das nicht falsch ist, so beinhaltet der Begriff etwas mehr. Zuerst erscheint eine Betrachtung der deutschen Fassung der ISO⁽³⁶⁾-Norm zum Begriff **Gebrauchstauglichkeit** von 1999 sinnvoll, die ein Jahr nach der Definition von Usability in der englischen ISO-Norm 9241-11 erschienen ist.

»Das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen.«

DIN EN ISO 9241-11: 1999, deutsche Übersetzung

Die Gebrauchstauglichkeit bezieht sich somit auf einen bestimmten **Nutzungskontext** und ist **relativ** zu den **Nutzungsanforderungen** zu sehen. Es geht folglich nicht nur um den Benutzer, der mit einem Arbeitsmittel adäquat umgehen muss, sondern auch um das Produkt selber, welches in einem **vorgegebenen Bezugsrahmen** die gestellten Anforderungen mit möglichst **geringem Aufwand, ohne Beeinträchtigungen** und mit **hinreichender Genauigkeit** so zu erfüllen hat, dass die Tauglichkeit des Produkts gegeben ist.

Begrifflichkeit

In dieser Arbeit wird entweder die englische Variante **Usability** oder der deutsche Begriff **Gebrauchstauglichkeit** verwendet. Synonyme wie »Handhabbarkeit« oder »User-Experience« treffen im Vergleich zum Begriff Gebrauchstauglichkeit inhaltlich nicht exakt das Ziel. Der im allgemeinen Sprachgebrauch verankerte Begriff »Benutzerfreundlichkeit« wird vom »Usability Kompetenzzentrum des Fraunhofer-Instituts für Ange-

(35) Der Begriff Produkt ist hier vorerst allgemein definiert und umfasst neben Hard- und Software auch Webpräsenzen und Dienstleistungen.

(36) **ISO** – International Organization for Standardization

wandte Informationstechnik« nicht empfohlen: »Benutzerfreundlichkeit suggeriert, dass es Menschen gibt, zu denen Produkte besonders nett sein müssen und dass diese Menschen empfindlich beziehungsweise nicht in der Lage sind, mit einem Produkt richtig umzugehen. Also spaltet der Begriff Benutzerfreundlichkeit die Welt in Menschen, die Produkte ohne Probleme nutzen können und Menschen, die Probleme bei der Nutzung von Produkten haben [FIT-USABILITY]«.

Usability noch neu?

Wenngleich Usability erst 1998 als ISO-Norm definiert wurde, so ist die zugrunde liegende Idee keineswegs mehr so jung. Bereits in den 1930er Jahren hat sich der franko-amerikanische Industriedesigner Raymond Loewy als Pionier auf diesem Gebiet intensiv mit Fragen zur Gebrauchstauglichkeit von Industrie-Produkten beschäftigt und einen ganzen Zoo an Maximen erdacht, wie Designer von ihrer Vorstellung eines »kompliziert-schmalzigen« Entwurfs ablassen und endlich die Zufriedenheit des Konsumenten anstelle der bloßen Optik in den Vordergrund stellen [RLOEWY].

Bedeutung der Gebrauchstauglichkeit für Anwender und Hersteller

Für **Anwender** geht es vor allem darum, möglichst zügig und unmissverständlich nachzuvollziehen, welchen Zweck das Produkt erfüllt, wo und wie es zu bedienen ist und oben-drein auch das Gefühl vermittelt zu bekommen, in den Bedürfnissen geachtet zu werden. Nur so erhält der Anwender Zugang zum Produkt und wird es auch öfter nutzen anstatt sich nach einer Alternative umzusehen.

So fördert z.B. das Vorhandensein eines **Wiedererkennungswerts** den Zugang – was ein Produkt durch Erfüllen von Konventionen leicht erreichen kann. Folgt das Produkt bestimmten Normen, so fördert dies deutlich die Zugänglichkeit, da vertraute Strukturen im Allgemeinen schnell erkannt werden und das Gefühl des Unbehagens bei Neuem mildern. Dieses Prinzip der gewohnten Struktur lässt sich leicht selber überprüfen wenn beispielsweise durch fremdsprachige Webseiten zu navigieren ist ohne mit der Sprache selber vertraut zu sein.

Für **Hersteller** ist der gebrauchstaugliche Entwurf ebenso wichtig. Um einen möglichst großen Kundenkreis an sich zu binden, versteht es sich von selbst, dass die Kommunikation zwischen Anwender und Hersteller auf Anwenderseite ein zufriedenes Gefühl der Fürsorge hinterlassen muss. Laut Stapelkamp muss die Usability fester Bestandteil der Corporate Identity sein, *»um für den Anwender ein markenkonformes Verhalten zu definieren, mit dem er an die Marke herangeführt wird und auf diesem Wege sowohl deren ideellen und funktionalen Nutzen, als auch deren offensichtlichen und verborgenen Vorteile erkennen kann [STAPELKAMP 2007]«*.

3.2 Zugänglichkeit von Web-Inhalten

Neben der Gebrauchstauglichkeit existiert ein weiterer Begriff, der zunehmend eine zentrale Bedeutung bei der Produktentwicklung erfährt: **Zugänglichkeit** als Übersetzung von **Accessibility**, im Deutschen besser bekannt als **Barrierefreiheit**, bezieht sich auf die uneingeschränkte Benutzung und richtet sich gezielt an Menschen, die in irgendeiner Form an der Nutzung eines Produkts gehindert sind. Dabei geht es nicht nur um Behinderungen, sprich körperliche oder geistige Einschränkungen, sondern um beliebige Umstände, die einen Menschen im Vergleich zu anderen Menschen an der aktiven Teilnahme eines Produkts hindern. Solche Barrieren können aufgrund von folgendem Kontext vorhanden sein [STAPELKAMP 2007].

- » Alter
- » Behinderung
- » Lernschwierigkeiten
- » Geschlecht
- » Sprache
- » sozio-kulturelle Bedingungen

Die Zugänglichkeit mit in diese Arbeit aufzunehmen hat zwei Gründe: zum einen gehören Gebrauchstauglichkeit und Zugänglichkeit zusammen, da beide, wenngleich sie auch auf verschiedenen Ebenen arbeiten, das gemeinsame Ziel der Produktverbesserung verfolgen.

Zum anderen wird für den Praxisteil dieser Arbeit eine Komponente⁽³⁷⁾ konzipiert, die – wenn auch mit Einschränkungen – eine Aussage zur Zugänglichkeit der Inhalte im Geobrowser macht. Dementsprechend wird das Thema Zugänglichkeit auf die Validierung von Web-Inhalte beschränkt. Weiterführende Themen wie gesetzliche Regelungen oder »assistive« (*unterstützende*) Technologien sind für diese Arbeit nicht relevant.

Um die Notwendigkeit zum barrierefreien Entwurf zu verdeutlichen, zeigt **Abbildung VI** den Anteil der US-amerikanischen Bevölkerung aus dem Jahr 2003, der aufgrund von Einschränkungen an der aktiven Teilnahme im Internet gehindert ist. Der prozentuale Anteil in der EU ist geringer – hier sind mit 8% ca. 38 Mio. Menschen betroffen [FRIEDMAN 2007].

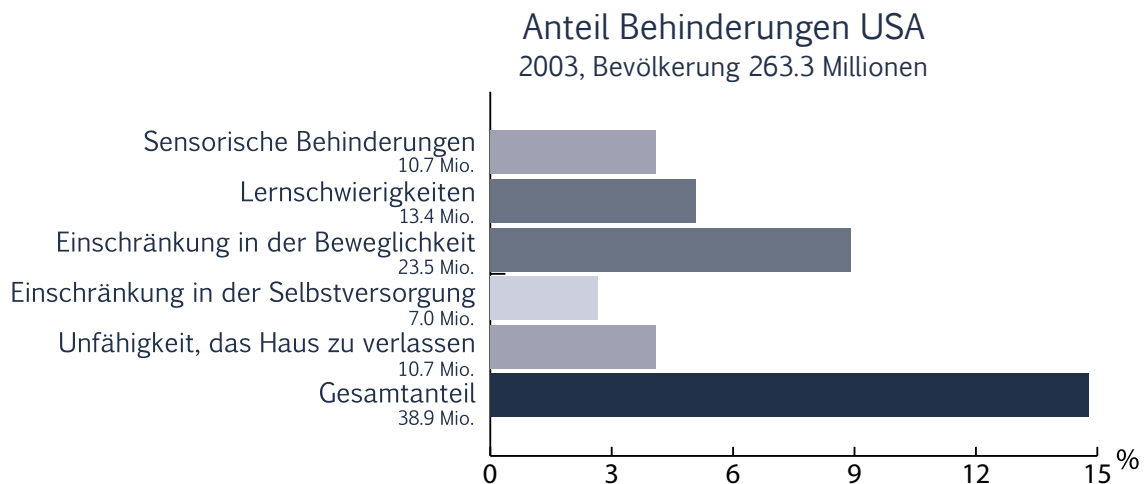


ABBILDUNG VI: Anteil Behinderungen USA 2003

Quelle: http://www.pascenter.org/state_based_stats/state_statistics_2003.php?state=us

Die Richtlinien zum barrierefreien Entwurf einer Webseite bilden die Standards des **W3C**⁽³⁸⁾. Diese bestehen hauptsächlich aus den folgenden zwei Aspekten:

- » **Barrierefreie Bedienbarkeit:** Die Bedienung einer Webseite muss auch mit der Tastatur zu vollziehen sein. Dabei muss das Markup semantisch korrekt und an »assistive« Technologien (*z.B. Bildschirmlesegeräte*) angepasst sein.

(37) Das Markup der KML-Ortsmarken sollte sich automatisch auf »Zugänglichkeit« validieren lassen.

(38) World Wide Web Consortium – ist das Gremium zur Standardisierung von Web-Techniken.

- » **Verständlicher und robuster Inhalt:** Neben klarer und korrekter Sprache muss die Funktionalität einer Seite für aktuelle und künftige Browser (*auch Textbrowser*) gewährleistet werden.

Die meisten der Anforderungen, besonders die Überprüfung des Markups, sind für ein rechnergestütztes Werkzeug zur Validierung von Webseiten leicht zu implementieren. Anforderungen zum klar verständlichen Inhalt lassen sich hingegen kaum automatisch validieren. Der **HTML-Validator**⁽³⁹⁾ als Dienst des W3C liefert gute Arbeit, um eine Webseite auf Einhaltung des Markups zu überprüfen.

Für Geobrowser gestaltet sich solch eine Validierung aufgrund der **Restriktion** des **Markups** und der **Entkopplung** vom gängigen **Webbrowser** etwas schwieriger. Dennoch können gewisse Kriterien untersucht werden, die als kurzer Report durchaus aufschlussreich sind und im Kapitel zur Konzeption wieder aufgegriffen werden. Dazu zählen z.B.:

- » Alternative Beschriftungen durch `alt`-Tags
- » Barrierefreie **Farbgestaltung**, d.h. ausreichende Farbkontraste
- » Verwendung von Navigationshilfen, z.B. `accesskey` oder `tabIndex`

3.3 Qualitätsmerkmale von Inhalten

Lässt sich die Qualität der Gebrauchstauglichkeit messen – oder anders gefragt, wie lässt sich objektiv ermitteln, ob ein Produkt gebrauchstauglich oder –untauglich ist? Ein untaugliches Produkt wird immer mit Mängeln einhergehen, die die Arbeit des Anwenders erschweren oder gar unmöglich machen. Das kann vieles beinhalten, z.B. *»unnötige und nicht für die Arbeitsaufgabe relevante Schritte, irreführende oder unzureichende Informationen, unerwartetes Systemverhalten oder Einschränkungen bei der Navigation* [PRO-CONTEXT]«. Ist damit aber ein Produkt, welches solche Mängel nicht aufweist, automatisch gebrauchstauglich? Und lässt sich das Maß an Gebrauchstauglichkeit messen und abstrakt darstellen, am besten mithilfe eines rechnerbasierten Werkzeugs?

(39) <http://validator.w3.org/>

Nach der ISO-Norm lauten die wichtigen qualitativen Kenngrößen **Effizienz**, **Effektivität** und **Zufriedenheit**, was zur Beurteilung immer das Feedback der Anwender impliziert. Dieses nun auf ein allgemein messbares System abzubilden gestaltet sich äußerst schwierig, da einerseits Anwender **unterschiedliche Wünsche** an ein Produkt stellen und genannte Kenngrößen ebenso vom **Bezugsrahmen** der Aufgabe **abhängen**.

Usability-Qualitätsfaktoren für Inhalte

Es ist an der Zeit, die Fokussierung von Produkte auf Inhalte zu spezialisieren, da dies dem eigentlichen Zweck dieser Arbeit entspricht. Und hier lassen sich die Kenngrößen der ISO-Norm so verfeinern, dass greifbare heuristische Faktoren entstehen, die die Überprüfung von Inhalten teilweise auch ohne Feedback ermöglichen und eine erste qualitative Aussage zur Gebrauchstauglichkeit liefern. **Vitaly Friedman** schreibt über Aspekte, die speziell die **Qualität** von **Web-Inhalten** bestimmen und somit unter dem Gesichtspunkt der Usability relevant sind [FRIEDMAN 2007].

- » **Konsistenz:** Sind alle Steuerelemente in ihrer Funktionalität inhaltsübergreifend?
- » **Logische Informationsstruktur:** Werden Informationen in einer übersichtlichen und klaren Struktur präsentiert?
- » **Kontextabhängige Orientierung:** Zeigen Inhalte dem Anwender stets, wo er sich gerade befindet, wo er war und wohin er gehen kann?
- » **Suchfunktion:** Zwingend notwendig für mittlere und große Auftritte.
- » **Navigation:** Existiert eine voraussehbare und verständliche Navigation?
- » **Qualität der Inhalte:** Sind ansprechende und sprachlich korrekte Inhalte vorhanden?

Jakob Nielsen als allgemein anerkannte Koryphäe zur Usability nennt mehrere **Qualitätskomponenten**, die meist auf den drei Kenngrößen der ISO-Norm aufsetzen und sich kaum objektiv überprüfen lassen, daher immer einen Probanden zur Beurteilung benötigen [NIELSEN].

- » **Lernbarkeit:** Wie leicht ist es für Anwender, grundlegende Aufgaben zu vollenden, wenn sie das erste Mal mit dem Design zu tun haben?
- » **Wiedererkennbarkeit:** Ist bei einem späteren Gebrauch der schnelle Wiedereinstieg in die Benutzung gewährleistet?

- » **Effizienz:** Wie schnell können Anwender Aufgaben erledigen, wenn sie das Design einmal kennen gelernt haben?
- » **Zufriedenheit:** Wie angenehm ist es, das Design zu verwenden?
- » **Fehlertoleranz:** Ist das Design resistent gegen falsche Nutzerführung?
- » **Verwendbarkeit:** Macht das Produkt das, was der Anwender braucht?

Zuletzt sei noch **Steve Krug** genannt, dessen erstes und wichtigstes Usability-Gebot *»Don't make me think!«* lautet. Die zugrunde liegende Erkenntnis besagt, dass Nutzer ungern ihre Zeit damit vergeuden, erst zu überlegen, wie etwas funktionieren könnte oder gemeint ist. Inhalte sollen selbsterklärend sein, ohne zusätzliche (*und meist unnötige*) Instruktionen und ohne dem Zuviel an nutzlosen Wörtern, so dass der *»Click«* ohne gedankliche Verzögerung erfolgt. Leicht verständliche Inhalte erfordern laut Krug **kein Nachdenken** und sind frei von **visuellem Rauschen** – womit eine visuelle Form von Lärm gemeint ist; zuviel um Aufmerksamkeit ringende Kaufeinladungen, Banner, Farben etc. [Krug 2002]. Zur Überprüfung der *»Krug'schen Gesetze«* ist wiederum das Feedback von Probanden gefordert.

Fazit

Zusammenfassend gilt, dass zur **qualitativen Untersuchung** von **Inhalten** eine Testreihe *ohne* Testpersonen, sei diese nun mithilfe eines Markup validierenden Werkzeugs oder subjektiv durch Qualitätsfaktoren überprüfende Experten durchgeführt, zum aktuellen Zeitpunkt nur eine eingeschränkte Aussagekraft besitzt. Für ein aussagekräftiges Urteil ist immer der *»Faktor Mensch«* vonnöten.

Für den Entwurf eines **softwarebasierten Werkzeugs** bedeutet dies, dass automatisch ablaufende Prozesse, wie die erwähnte Validierung des Markups, mit menschlich erzeugten Daten, d.h. durch Usability-Tests erhaltene Ergebnisse, kombiniert werden müssen. Ein ganzheitliches Werkzeug sollte demnach Inhalte nicht nur auf die **Einhaltung von Konventionen** hin überprüfen, sondern auch in der Lage sein, die Aktionen der Probanden in der einen oder anderen Form mit in die Untersuchung mit einzubeziehen. Durch diese Kombination wird es mit dem Werkzeug möglich sein, die Gebrauchstauglichkeit von Inhalten objektiv zu messen und sie visuell aufbereitet den Anwendern als Ergebnis darzustellen.

3.4 Untersuchungsverfahren für Inhalte

Da erläutert wurde, bei welchen Merkmalen anzusetzen ist, um die Gebrauchstauglichkeit zu untersuchen, sollen nun gängige **Analyseverfahren** aus der Praxis betrachtet werden. Was hat es generell mit dem Begriff Analyse auf sich?

Analyse

»Eine Analyse ist eine ganzheitliche, systematische Untersuchung, bei der das untersuchte Objekt oder Subjekt in seine Bestandteile zerlegt wird und diese anschließend geordnet, untersucht und ausgewertet werden. Dabei dürfen die Vernetzung der einzelnen Elemente und deren Integration nicht außer Acht gelassen werden.«

Quelle: Wikipedia [<http://de.wikipedia.org/wiki/Analyse>]

Ein zu untersuchendes Problem wird zuerst immer in **Teilprobleme** zerlegt, die einzeln beobachtet und untersucht werden. Ist dies geschehen und Erkenntnis über die Struktur gewonnen, lassen sich die einzelnen Faktoren wieder zusammensetzen, um so Erkenntnis über das Ganze zu erhalten. Natürlich ist es von Relevanz, was überhaupt untersucht werden soll. Die Beobachtung eines physikalischen Vorgangs ist nicht gleichzusetzen mit dem Beobachten eines Probanden, der gerade im Browser navigiert. Wenn Erstgenanntes bei jedem Test immer dasselbe Ergebnis liefert, so lässt sich daraus meist direkt ein Erkenntnis oder besser gesagt ein Gesetz ableiten. Beim Probanden ist dies nicht zwangsläufig der Fall, da die Interpretation des Verhaltens immer subjektiv ist.

Gängige Analyseverfahren

In der Usability-Forschung wird primär unterschieden zwischen **qualitativen** und **quantitativen** sowie **heuristischen Analyseverfahren** [STAPELKAMP2007].

Dabei basieren **quantitative Verfahren** immer auf Testreihen, die sich in irgendeiner Form statistisch darstellen lassen. Gängiges Beispiel sind **Fragebögen**, die einer ausreichend großen Menge an Probanden zur Beantwortung übergeben werden. Dieses aus der Marktforschung bekannte Prinzip hat sich in der Usability durchgesetzt und beinhaltet

meist, die Probanden im Anschluss einer Sitzung die Inhalte nach mehreren Kriterien, wie z.B. **Navigation** oder **Verständlichkeit**, bewerten zu lassen. Die Ergebnisse lassen sich sehr gut in Diagrammform darstellen und haben ab einer ausreichenden Probandenzahl eine gewisse Aussagekraft, da erkennbar wird, wo eklatante Schwächen liegen.

Qualitative Verfahren gehen meist tiefer in die Materie und bezeichnen **Interviews** als Einzelgespräch oder Gruppendiskussion, die auf den Erfahrungen und Beobachtungen eines Testablaufs aufsetzen, so dass Testleiter und Probanden gemeinsam über Positives und Negatives diskutieren können.

Heuristische Verfahren basieren auf den genannten Qualitätsmerkmalen und werden in der Regel durch Experten anstelle von Testpersonen durchgeführt. Im Sinne der **Heuristik** (*Kunst der Wahrheitsfindung*) werden erprobte Muster, ungeschriebene Gesetze und Prinzipien angewandt, um Inhalte auf offensichtliche Mängel zu untersuchen. Erfahrene Experten erkennen meist sofort, ob Steuerelemente, Kommunikation und Navigation einer erprobten inneren Struktur folgen. Somit ist solch eine Untersuchung insbesondere vor einem Probandentest sinnvoll, um ein Testergebnis nicht mit der Entdeckung offensichtlicher Mängel zusätzlich zu beschweren.

Tracking als Spezialisierung quantitativer Verfahren

Für diese Arbeit relevant sind insbesondere sog. **Tracking-Verfahren**, die als Untermenge der quantitativen Verfahren immer ein Werkzeug benötigen, welches das Nutzerverhalten in irgendeiner Form aufzeichnet und ebenso in der Lage ist, dieses auch wiederzugeben.

Screen-Capturing beschreibt das Aufzeichnen des aktuellen Bildschirminhalts mit anschließendem Export als Video-Datei. Dies ist eines der einfachsten Mittel, um eine Webseite zu untersuchen. Als Ergebnis erhält man allerdings auch nur eine unkommentierte Wiedergabe des Testablaufs. Das Produkt mit der größten Verbreitung und Akzeptanz ist »**Camtasia Studio**«⁽⁴⁰⁾, wo eine Lizenz ab 200€ zu haben ist.

- + mit der nötigen Software **leicht** und **kostengünstig** zu implementieren
- + Mauszeiger kann Aufschluss über **Navigationslücken** geben

(40) <http://www.techsmith.de/camtasia/features.asp>

- **subjektive Interpretation** des Materials
- **Aussagekraft** als alleiniges Verfahren stark **eingeschränkt**
- bei einer großen Menge an Probanden mit **hohem Speicherbedarf** verbunden

Als begleitende Maßnahme zum Screen-Capturing kommen oft **Videoaufnahmen des Probanden** hinzu: Mimik und Gestik können möglicherweise Aufschluss darüber geben, *warum* der Mauszeiger besonders oft auf ein Bildelement zeigt.

- + In Kombination mit Screen-Capturing gewisse **Aussagekraft**
- **subjektive Deutung**
- **hoher Kosten- und Konfigurationsaufwand** für entsprechende Hard- und Software

Als zusätzliches Mittel werden Probanden oft zum **lauten Denken** aufgefordert, um Auskunft über ihr momentanes Handeln und Empfinden zu geben. Diese **Sound-Aufnahmen** werden meist in Kombination zu Screen-Capturing und Videoaufnahmen eingesetzt.

- + In Kombination mit Screen-Capturing und Videoaufnahmen **aussagekräftig**
- + durch Probandenfeedback **Deutung nicht mehr subjektiv**
- Nur in **Kombination** mit anderen Methoden **sinnvoll**
- **entsprechende Hardware** muss **vorhanden** sein

Unter **Eye-Tracking** versteht man nichts anderes als das Erfassen des Blickwinkels des Probanden mithilfe entsprechender Hard- und Software. So wird während eines Tests die Augenbewegung des Benutzers aufgezeichnet und später auf das aufgezeichnete Material gelegt. Erfassen lässt sich neben dem Winkel auch die Blickdauer, die sich einem bestimmten Bildelement zuordnen lässt.

- + **aussagekräftig**, wenn wichtige Inhalte vom Probanden nicht wahrgenommen wurden
- + kann darlegen, ob Nutzer sich auf Inhalte **konzentrieren** oder sie nur **überfliegen**
- + **Veränderungen des Pupillendurchmessers** geben Auskunft, ob unbekannte bzw. irrelevante oder erwartete Begriffe und Bereiche erfasst wurden
- **kostenintensiv** und **zeitintensive Konfiguration**
- **Ergebnisse** teilweise **ungenau**, realer Blickpunkt nicht genau abgebildet

Ergebnispräsentation und Berichterstattung

Die Möglichkeiten zur Dokumentation einer Usability-Evaluation sind vielfältig. In der Usability-Szene hat sich inzwischen das **Common Industry Format** (CIF)⁽⁴¹⁾ für Usability-Tests durchgesetzt, welches vom **National Institute of Standards and Technology**⁽⁴²⁾ (NIST) definiert wurde. Hierbei handelt es sich um ein Template für die Berichterstattung von Usability-Tests. Das **NIST** schlägt vor, dass im Bericht Statistiken über die in der **ISO Norm 9241** festgehaltenen Qualitätsmerkmale aufgeführt werden [USABILITY-NEWS-CH].

- » Kann der Anwender seine Aufgabe zufrieden stellend lösen? (**Effektivität**)
- » Wie lange brauchte der Anwender, um dies zu tun? (**Effizienz**)
- » Waren die Anwender zufrieden? (**Zufriedenheit**)

Zudem sollte ein Testbericht unter anderem Folgendes beinhalten:

- » **Ziel und Zweck** – warum wurde der Usability-Tests durchgeführt?
- » Klare **Produktbeschreibung**
- » **Beschreibung** der beabsichtigten **Anwender**
- » **Experimentelles Setting** – z.B. unter Realbedingungen oder im Usability-Labor
- » Zugrundeliegende **Aufgabenstellungen**
- » **Anzahl** an **Testteilnehmern** sowie **Kriterien** zur Auswahl der Teilnehmer
- » **Methode** und der **Prozess**, die dem Test als Basis zugrunde lagen

3.5 Werkzeuge für Web-Inhalte

Wie deutlich wurde, gibt es aktuell *kein* Usability-Universalwerkzeug, sondern nur die Möglichkeit, je nach Budget, mehr oder minder umfangreiche Tests in Usability-Laboren durchführen zu lassen. Dieses wird in der Regel von mittleren und großen Unternehmen auch wahrgenommen, da zeitliche und finanzielle Kapazitäten vorhanden sind. Was aber ist mit Kleinunternehmen oder Privatpersonen, die nur schnell und kostengünstig ihre Web-Inhalte untersuchen und verbessern wollen?

(41) siehe dazu <http://zing.ncsl.nist.gov/iusr/documents/cifv1.1b.htm>

(42) <http://www.nist.gov/>

Für diesen Bedarf sind in letzter Zeit etliche freie und kostenpflichtige **Web-Tracking-Werkzeuge** erschienen, die versuchen, das Verhalten der Besucher dem Webseitenbetreiber transparent zu machen: woher kommen Besucher, wie haben sie die Seite gefunden, wie lang bleiben sie und wohin klicken sie wie oft? Diese Daten sind nur **teilweise Usability-relevant** und zielen mehr aufs **Marketing**. Dennoch können Daten wie Klickverhalten oder Verweildauer auch Aufschluss über Navigationslücken geben.

Ein junges Medium zur Visualisierung des Nutzerverhaltens sind so genannte **Heatmaps**, mit denen die **zweidimensionale visuelle Darstellung** der Nutzeraktionen⁽⁴³⁾ bezeichnet wird. Meist dient ein abgedunkelter Screenshot der Webseite als Hintergrund, auf dem die Daten je nach Häufigkeit des Auftretens mit unterschiedlichen Wärmegraden eingefärbt werden. So ist sofort zu erkennen, wo auf einer Webseite viel Nutzeraktivität stattfindet und wo nicht. **Abbildung VII** zeigt eine prinzipielle Heatmap.



ABBILDUNG VII: Prinzipieller Aufbau einer Heatmap

(43) Bei Webdiensten beinhalten Nutzeraktionen nur Mausklicks und Mausbewegungen. In Verbindung mit Eye-Tracking Systemen kann eine Heatmap auch die Blickverläufe und die Blickdauer abbilden.

Programme, die dieses leisten, sind z.B. **Clicktracks Appetizer**⁽⁴⁴⁾ oder **CrayzEgg**⁽⁴⁵⁾. In der neuesten Version unterstützt auch **Google Analytics**⁽⁴⁶⁾ die Darstellung von Heatmaps.

3.6 Ansätze für Geobrowser

Für die Untersuchung der Inhalte im Geobrowser kommen generell alle Verfahren infrage, die auch für Web-Inhalte gelten. Auch wenn es sich beim Erstellen von Inhalten nicht um **klassische Webgestaltung** handelt, sondern aufgrund der stärkeren **multimedialen Betonung** eher um **Inszenierungen**, so ist das Prinzip der Aufbereitung dennoch mit dem von Webseiten zu vergleichen. Was ein Werkzeug für Geobrowser anbelangt – und im Rahmen dieser Arbeit geht es um ein **softwarebasiertes Werkzeug** –, so ist es ebenso von Relevanz, dass momentan viele Unternehmen, die Inhalte für Geobrowser veröffentlichen, junge Start-Up-Unternehmen sind, die in den seltensten Fällen Kapazitäten für Tests an Usability-Laboren zur Verfügung stellen können – dafür aber einen Bedarf an einem **kostengünstigen** und **leicht einsetzbaren Werkzeug** haben.

Qualitative Verfahren lassen sich aufgrund der Beschränktheit auf die Mensch-Mensch Interaktion kaum mit einem Werkzeug vereinen, **quantitative Verfahren** sehr wohl. Ein softwarebasiertes Werkzeug sollte im Sinne quantitativer Verfahren neben **Tracking-Funktionen** auch die Möglichkeit bieten, die Definition, Auswertung und Präsentation von **Fragebögen** (Probandenfeedback) zu vereinfachen.

Tracking-Verfahren, die visuell aufbereitet die Nutzeraktivität wiedergeben, sollten Ressourcen schonen und gleichzeitig genau das können, was Aufnahmeverfahren wie Screen-Capturing auch leisten. **Heatmaps**, die bislang im Webbrowser in zweidimensionaler Form existieren, sollten **dreidimensional** im **Geobrowser** ihr Korrelat finden. Andere Verfahren wie Eye-Tracking oder Video- und Audioaufnahmen vertragen sich nicht mit einem leicht zu implementierenden und kostengünstigen Werkzeug.

(44) http://www.clicktracks.com/products/appetizer/appetizer_test_with_avinash.php

(45) <http://crazyegg.com/>

(46) <http://www.google.com/analytics/de-DE/>

Zuletzt bleiben die **heuristischen Verfahren**, die sich ebenso übertragen lassen – dabei allerdings modifiziert werden müssten. Hier besteht die Möglichkeit, Inhalte für Geobrowser bezüglich der Zugänglichkeit zu validieren und ebenso gewisse erprobte **Muster** und **Konventionen** bei Geobrowser-Inhalten zu überprüfen – solche, die sich bei Web-Inhalten bewährt haben und für Geobrowser-Inhalte angepasst werden können, und auch solche, die sich direkt aus den Gegebenheiten des Geobrowsers ableiten lassen.

Ein neuer bislang unerprobter Ansatz ist bei einem softwarebasierten Werkzeug die automatische Evaluation, die im Folgenden als **auto-evaluierendes Verfahren** bezeichnet und eingeführt wird. Die Idee dahinter ist, dass ab einer ausreichend großen Menge an Nutzerdaten zu einer Sitzung es möglich ist, diese Daten in **Klassen ähnlicher Ergebnisse** zu unterteilen. Geringe Abweichungen werden dabei nicht in Betracht gezogen. In Kombination mit den Ergebnissen der anderen Verfahren soll automatisch ein **Report** erstellt werden, möglichst auf Basis des **Common Industry Formats**, der Analysten eine schnelle Übersicht der Ergebnisse sowie eine automatisch erstellte Bewertung des KML-Materials liefert. Im Kapitel zur Konzeption wird dieser Ansatz genauer beschrieben, hier soll er erst einmal nur genannt werden.

Fazit

Ein softwarebasiertes Werkzeug zur Untersuchung von Geobrowser-Inhalten sollte **leicht zu erlernen, kostengünstig, ohne viel Aufwand** betrieben werden können und die infrage kommenden Usability-Verfahren realisieren. Mit dem **geolyzer** existiert bereits ein Prototyp, der sich gut eignet, die genannten Ansätze für den Bereich der Geobrowser umzusetzen. Im nächsten Kapitel wird der Ist-Zustand dieses Prototyps genauer beschrieben und ein Soll-Zustand für das Endprodukt definiert.

“A journey of a thousand miles begins
with a single step.”

Laozi
Philosopher of Ancient China

4. geolyzer als erster Ansatz

Im Mai 2008 war es an der Zeit, den geolyzer einem ersten Nutzertest zu unterziehen. So wurde eine freiwillige Testperson kurzerhand vor das Programm gesetzt und zum Testen beordert. Einen Tag vorher hatte ich den geolyzer noch komplett inspiert und bis auf wenige Holprigkeiten abgesegnet. Aber es kam anders, und um es kurz zu machen wuchs die Erkenntnis, dass das Programm zur Untersuchung und Verbesserung der Usability selber noch deutliche Navigationslücken aufwies. Vielleicht hätte parallel ein Programm entwickelt werden sollen, welches speziell und objektiv die Gebrauchstauglichkeit des geolyzers überprüft?

Sei es drum, inzwischen taugt auch der geolyzer mehr zum Gebrauch und soll in seiner Konstruktion vorgestellt werden. Zuerst wird auf Historie, Funktionalität, Technologie und Architektur des Ist-Zustandes eingegangen. Bei den Technologien wird der Datenbankzugriff via ADO.NET sowie die serverseitige Programmierung von Funktionen und gespeicherten Prozeduren mittels Transact-SQL genannt. Aber auch das Thema zum Entwurf von Benutzersteuerelementen soll nicht fehlen. Abschließend wird der Soll-Zustand für das Endprodukt definiert.

4.1 Historie

Das Grundkonzept eines softwarebasierten Werkzeugs zur Untersuchung von Geobrowser-Inhalten wurde im Rahmen des Kurses »Webpräsenzen in Geobrowsern« an der Hochschule Bremen zwischen Januar und Februar 2008 erdacht. Ende Februar 2008 entstand ein erster Prototyp mit folgender Minifunktionalität:

- » **Integration** des **Google Earth-Client** in eine C#-Windows-Anwendung über COM
- » **Verbindung** zum **SQL Server** mittels ADO.NET
- » **Persistente Speicherung** der Google Earth **Kamerapositionen**

Im Folgekurs »Applikationsentwicklung in Geobrowsern« wurde dieser Prototyp zwischen April und Juni 2008 so erweitert, dass eine Software entstand, die als Grundgerüst der zu entwickelnden Usability-Komponenten dient. Diese Erweiterung realisiert **Verwaltungsfunktionen** für Sitzungen und Probanden, Abbildung der wichtigsten **Google Earth Funktionen** auf die Anwendung, ein **Software-Setup** sowie angemessenes **Design** und **Layout**.

Abbildung VIII zeigt den Status quo der drei Meilensteine. Zum ersten Meilenstein im Februar 2008 existierten vier Klassenpakete und am bislang letzten Meilenstein vom 1. Juli 2008 wurde mit 67 Klassen ca. **65%** der erwünschten **Funktionalität** realisiert.

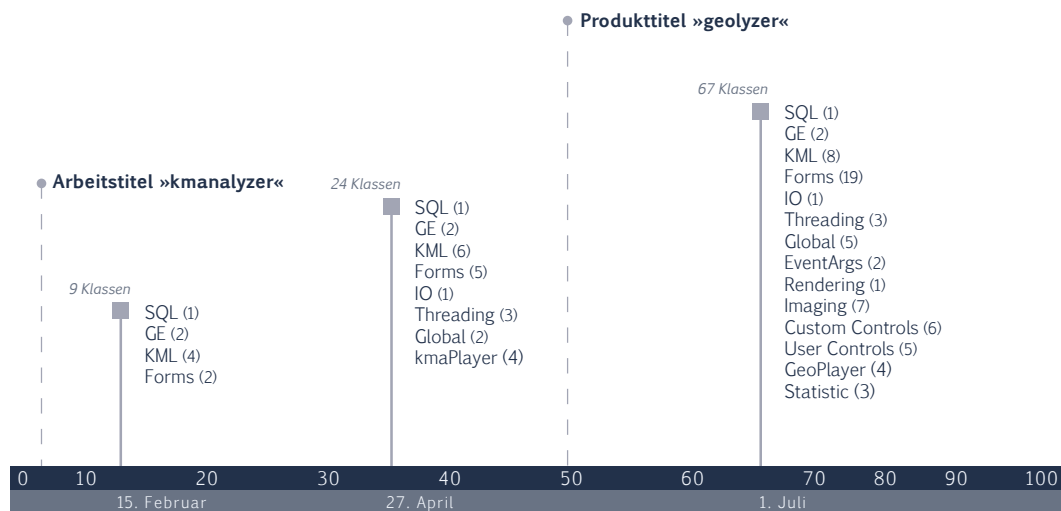


ABBILDUNG VIII: Historie geolyzer

4.2 Funktionalität

Das Prinzip des **geolyzer** lässt sich mit wenigen Worten beschreiben: Analysten legen eine **Sitzung** an, weisen dieser **KML-Daten** zu, definieren **Aufgaben** sowie **erwünschtes Feedback** und starten für Probanden eine **Aufnahme**, wo diese durch Inhalte **navigieren**, **Aufgaben** beachten und im Anschluss der Aufnahme ein **Feedback** abgeben. Jegliche relevante **Nutzeraktivität**⁽⁴⁷⁾ wird persistent gespeichert. **Abbildung IX** zeigt das entsprechende Anwendungsfalldiagramm.

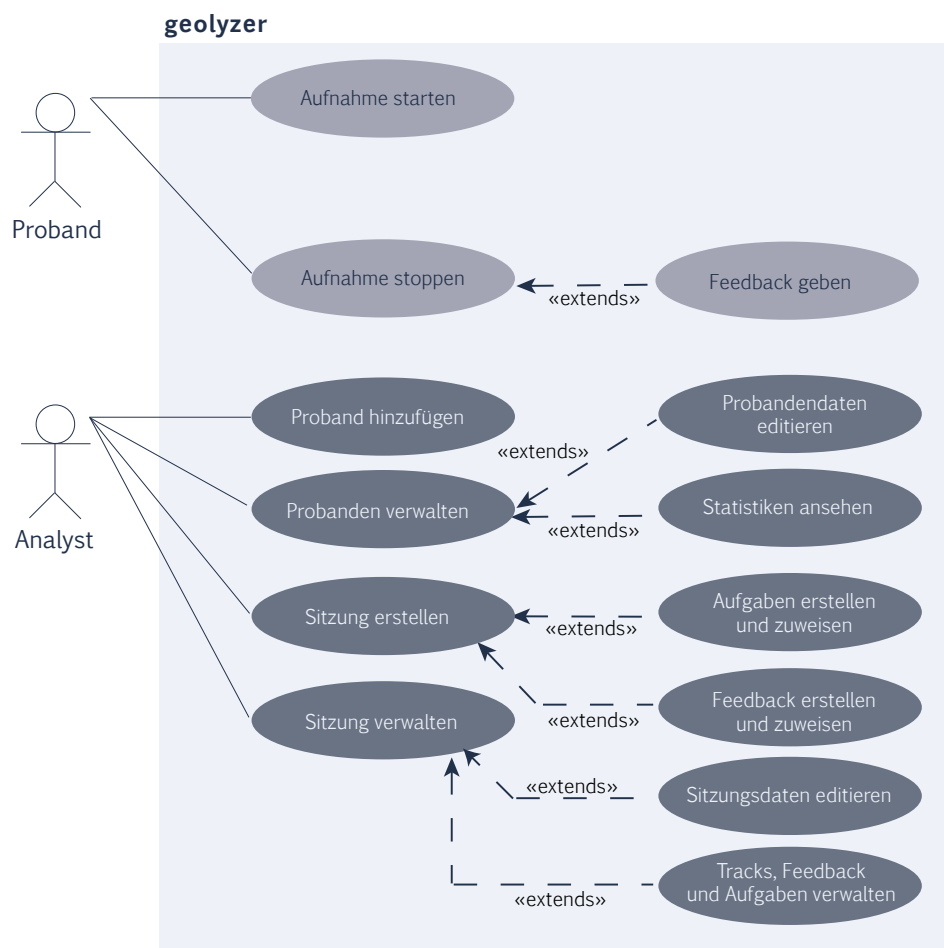


ABBILDUNG IX: Anwendungsfalldiagramm Ist-Zustand

(47) Dabei handelt es sich nicht nur um den Kamerafokus sondern auch um Nutzerereignisse wie Suche, Ebenen ein- und ausblenden sowie das Anwählen von Ortsmarken.

Mit der Möglichkeit für Analysten, einen **Fragenkatalog** nach eigenen Vorstellungen für Probanden zu definieren und diesen im Anschluss einer Aufnahme von Probanden beantworten zu lassen, sind bereits erste Ansätze **quantitativer Usability-Verfahren** implementiert. Allerdings existiert noch keinerlei Funktionalität, die dieses automatisch auswertet, in irgendeiner Form visualisiert oder als Report exportiert.

Damit ist die Grundfunktionalität gegeben. Ein ausführliches Benutzerhandbuch ist im laufenden Programm unter dem Menüpunkt »Hilfe« abrufbar. Adäquate Informationen zur Installation liefert das Setup-Programm der Installations-CD.

4.3 Technologie

Beim **geolyzer** kommt fast ausschließlich **Microsoft-Technologie** zum Einsatz. Einer der Gründe dafür sind die an der Hochschule Bremen vorhandenen Lizenzen, ein weiterer Grund ist die Zuverlässigkeit und der Komfort aktueller Microsoft-Entwicklungsumgebungen. Folgende Werkzeuge und Technologien sind relevant:

- » Microsoft **Visual Studio 2008** als Entwicklungsumgebung
- » Microsoft **SQL Server 2005** Standard, Express und Developer Edition
- » ADO.NET
- » Transact SQL
- » XML

Um Anwendern von **geolyzer** keine zusätzlichen Anschaffungskosten aufzubürden, wird auch die kostenfreie **SQL Server 2005 Express-Version** unterstützt. Diese hat im Vergleich zu früheren kostenfreien Versionen deutlich weniger Einschränkungen. Eine maximale Datenbankgröße von **4 GB** im Vergleich zu **1 Exabyte (1 Mio. Terabyte)** bei der **Standard Edition** sollte für **geolyzer** selbst bei sehr umfangreichen Usability-Tests kein Problem darstellen. Die Beschränkung auf eine einzige CPU allerdings könnte bei mehreren Clients im Rechnernetz zu Verzögerungen führen. Daher ergibt es Sinn, das fertig entwickelte Produkt besonders mit der **Express Edition** im **belasteten Rechnernetz** zu testen.

4.3.1 ADO.NET 2.0

Will eine laufende Anwendung auf eine Datenquelle zugreifen, so geschieht dies in der Regel mit einem **ODBC-Treiber**⁽⁴⁸⁾. Seit dem .NET-Framework 2.0 ist mit der zweischichtigen Klassenhierarchie **ADO.NET 2.0** (*ActiveX Data Objects .NET*) eine komfortable Technologie vorhanden, mit der auf Datenquellen zugegriffen werden kann. Für den schnellen, schreibgeschützten Zugriff auf Daten wurden Komponenten entworfen, die als Objekte direkt mit der Datenbank verbunden sind. Diese von Microsoft als **Datenanbieter** bezeichneten Klassen ermöglichen zusammen mit **datenquellenunabhängigen Objekten** eine für Entwickler leicht zu implementierende, zweischichtige Klassenstruktur zur Kommunikation zwischen Anwendungsobjekten und Datenbank. **Abbildung X** zeigt die Architektur von ADO.NET [MSDN LIBRARY].

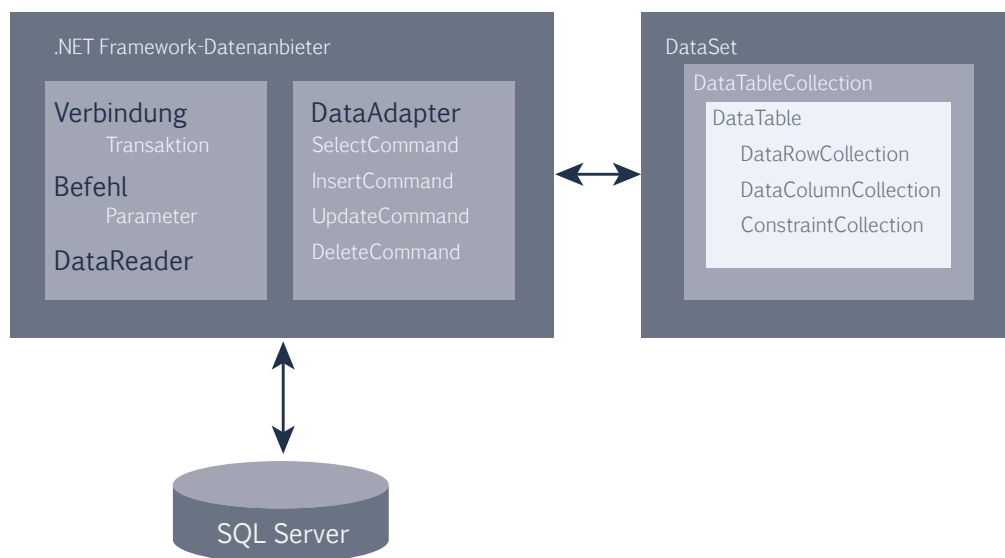


ABBILDUNG X: Architektur ADO.NET, Quelle: MSDN Library

Um Daten vom SQL Server abzufragen und diese mit Objekten der laufenden Anwendung interagieren zu lassen, existieren die datenquellenunabhängigen Container `DataSet` und `DataTable`, wobei das `DataSet` eine Kollektion für `DataTable`-Objekte ist, mit der sich theoretisch ein temporäres Abbild der gesamten Datenbank-Tabellen im lokalen Speicher erzeugen lässt.

(48) Open Database Connectivity – Standardisierte Methode, die den Zugriff auf Datenbanken erlaubt, ohne dabei zu berücksichtigen, aus welchem Programm oder von welchem Betriebssystem aus der Zugriff erfolgt.

Ein Vorteil eines `DataTable`-Objekts ist es, diese Datenstruktur direkt an ein **Steuer-element** zu **binden**. So lässt sich beispielsweise mit einem einzigen Abfrage-Statement, welches aus mehreren `SELECT`-Anweisungen besteht, ein `DataSet` mit mehreren Tabellen füllen, die sich in die Kollektion als `DataTable`-Objekte einreihen. Und diese werden an die `DataSource`-Eigenschaft mehrerer »*ComboBoxes*« gebunden – kein manuelles Auffüllen von Daten und ein Vorgang, der im **geolyzer** oft Verwendung findet.

Eine weitaus mächtigere Eigenschaft der `DataTable`-Datenstruktur offenbart sich, wenn diese an die .NET 2.0 Repräsentation einer Tabelle – ein Objekt der Klasse `DataGridView` – gebunden wird. Nutzer können in der Tabelle Einträge ändern und das gebundene `DataTable`-Objekt kann später zum automatischen Update der geänderten Daten en bloc mit der Datenquelle veranlasst werden. Der Komfort der Lösung täuscht etwas über den notwendigen Programmieraufwand hinweg.

Um überhaupt Verbindung zu einem Datenbankserver aufzubauen, ist der Namensraum `System.Data` sowie der entsprechende Datenprovider – in diesem Fall `System.Data.SqlClient` – über die `using`-Direktive⁽⁴⁹⁾ einzubinden. Zur Herstellung der Kommunikation sind wenige Objekte vonnöten:

- » **SqlConnection** – Verbindungsaufbau via Verbindungszeichenfolge. Enthält den Instanznamen des SQL-Servers und weiterhin Authentifizierungsinformationen
- » **SqlDataAdapter** – Brücke zwischen Datenquelle und `DataSet`
- » **SqlCommand** – Zugriff auf gespeicherte Prozeduren sowie parametrisierte Queries
- » **SqlParameter** – stellt Parameter für `SqlCommand` dar. Damit lassen sich beispielsweise Binärdaten-Abfragen realisieren

(49) Die `using`-Direktive zum Importieren von Typen ist nicht zu verwechseln mit der `using`-Anweisung, die die richtige Verwendung von `IDisposable`-Objekten sicherstellt. Dadurch lässt sich der Lebenszyklus von Objekten auf einen sog. `using`-Block beschränken. Nach verlassen des Blocks werden die Objekte sofort aus dem Speicher entfernt.

4.3.2 Transact-SQL

Im Bereich der relationalen Datenbankenmanagementsysteme ist **SQL**⁽⁵⁰⁾ als Datenbanksprache nicht wegzudenken. Seit dem ersten **ANSI**⁽⁵¹⁾ Standard von 1986 wurden viele Erweiterungen hinzugefügt und fast ebenso viele Dialekte kamen hinzu. Das liegt vor allem daran, dass die grundlegenden Befehle von SQL zwar schön und gut sind, aber die meisten Hersteller für ihre Produkte den Funktionsumfang teils drastisch erweitert haben.

So tat es auch Microsoft und führte mit **Transact-SQL** einen Dialekt ein, der für SQL-Kunde eines anderen Systems mit einem Einarbeitungsaufwand verbunden ist. Das beginnt bei der Bezeichnung der grundlegenden Datentypen und macht auch bei gängigen Sprachelementen nicht halt. Typen wie `money`, `xml` oder `bit` findet man bei anderen Dialekten gar nicht oder unter einem anderen Bezeichner, das automatische Inkrementieren eines künstlichen Primärschlüssels vollzieht sich durch das Schlüsselwort `IDENTITY` und das Erstellen eines spezifischen Installationsskripts mit Nutzern und Rechten ist eine Kunst für sich.

Für eine Software wie **geolyzer** ist es nur naheliegend, so viel Anwendungslogik wie möglich an den Server auszulagern, denn bei einem Usability-Test kommunizieren **geolyzer**-Clients im Rechnernetz mit einem oder mehreren SQL-Servern und tauschen ständig Daten aus. Um das Netz mit zum Teil sehr langen `SELECT`-Statements nicht unnötig zu belasten und wiederkehrende Abläufe einmal zentral an nur einer Stelle zu verwalten, existieren **gespeicherte Funktionen** (*engl. User Defined Functions*) und **gespeicherte Prozeduren** (*engl. Stored Procedures*).

Die Unterschiede liegen vor allem darin, dass gespeicherte Prozeduren zum **Anlegen**, **Löschen** und **Ändern** von Daten verwendet werden und Funktionen eher zum **Zugriff** auf Daten fungieren. Auch wenn gespeicherte Prozeduren über den Ausgabeparameter `OUTPUT` einen Wert zurückgeben können, so ist das Zurückgeben einer ganzen Tabelle nur durch eine Funktion zu realisieren [MERTINS 2006].

(50) **SQL** – Structured Query Language

(51) **ANSI** – American National Standards Institute

4.3.3 Benutzersteuerelemente in .NET

Das .NET-Framework beinhaltet eine reichhaltige Auswahl an Steuerelementen, mit denen sich die Anforderungen der meisten Anwendungen erfüllen lassen. Soviel zur Theorie. In der Praxis kommt es allerdings nur zu oft vor, dass eine erforderliche Funktionalität nicht vorhanden ist, geändert werden muss oder es einer an einen gestalterischen Entwurf angepassten Optik bedarf.

Eine der größeren Herausforderungen beim Entwickeln von Windows Anwendungen bleibt das Entwickeln eigener Steuerelemente. Um tiefer zu verstehen, was der Begriff **Benutzersteuerelement** beinhalten kann, werden die im Amerikanischen geläufigen Differenzierungen betrachtet [MACDONALD 2006].

- » **Custom Control** – allgemeiner Ausdruck für Benutzersteuerelemente
- » **User Control** – Komposition an Steuerelementen
- » **Owner-Drawn Control** – Steuerelemente mit überschriebener Zeichnungslogik
- » **Derived Control** – von einem anderem Steuerelement abgeleitetes Element

Custom Control

Dieser Ausdruck bezeichnet als Oberbegriff ein in irgendeiner Form selbst erstelltes oder angepasstes Steuerelement. Er kann analog zum deutschen Ausdruck »Benutzersteuerelement« verwendet werden.

User Control

Hiermit wird ein spezielles .NET Feature bezeichnet. Ein User Control ist eine **Komposition** aus mehreren bereits **existierenden** oder **selbst erstellten Steuerelementen**. Einfaches Beispiel

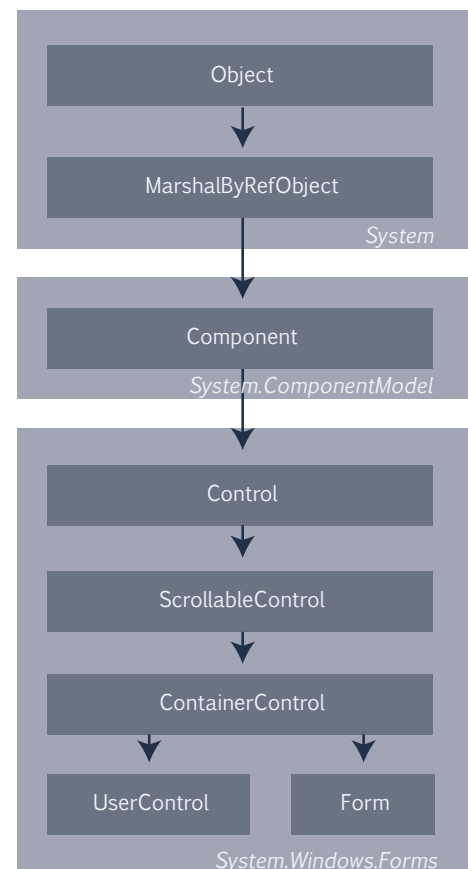


ABBILDUNG XI:
Windows Control Hierarchie [MACDONALD 2006]

wäre eine Eingabemaske, die aus mehreren Textfeldern besteht. Diese Komponente könnte zusätzliche Logik zum Validieren der Textfeld-Inhalte implementieren und als Klassenbibliothek veröffentlicht werden.

Derived Control

Bezeichnet ein von einem bereits vorhandenen Steuerelement **abgeleitetes Steuerelement**, welches Funktionalität überschreibt und/oder hinzufügt. Meist ist dies der erste Ansatz beim Entwurf eines eigenen Steuerelements. Es wird ein passendes Basiselement zum Ableiten ausgewählt, welches bereits eine gewisse Funktionalität implementiert, die nicht mehr selber entworfen und getestet werden muss.

Owner Drawn Control

Soll eine Anwendung nicht wie eine Windows Anwendung aussehen, so bleibt Entwicklern nichts anderes übrig als die **Logik zum Zeichnen** zu **überschreiben** und mit den Mitteln von **GDI+**⁽⁵²⁾ selber zu implementieren. Dieser Prozess kann sich als sehr aufwändig gestalten, da die Logik zum Zeichnen bei vielen Windows Steuerelementen sehr tief in der Hierarchie angesiedelt ist⁽⁵³⁾. Im optimalen Fall muss nur die `OnPaint`-Methode eines Steuerelements überschrieben werden.

Beim **geolyzer** kommen alle Arten von **Custom Controls** vor. Um dies an einem Beispiel zu demonstrieren, sei der **geoPlayer** genannt. Dieser wurde als **User Control** realisiert. Als Komposition beinhaltet er mehrere Steuerelemente, darunter Standard Windows Steuerelemente – z.B. `Label` für Track- und Zeitanzeige – sowie **Derived** und **Owner-Drawn Controls**. Die Suchleiste ist ein Beispiel für ein **Derived Control**. Dieses ist von der Bedienung fast identisch zum *Windows Media Player*, d.h. der Nutzer kann durch Ziehen der Suchleiste im Track problemlos navigieren. Sie wurde von der `Panel`-Klasse abgeleitet, die bereits eine Grundfunktionalität als Container-Element realisiert. Nichtsdestotrotz musste ein Großteil der Funktionalität, z.B. Drag-and-Drop und Maus-Ereignisse, manuell implementiert werden. Die Suchleiste ist aber auch gleichzeitig ein **Owner-Drawn Control**, da die `OnPaint`-Methode des inhaltlosen `Panel`s überschrieben wurde, um die abgelaufene Zeit sowie den Schieberegler gemäß dem aktuellen Zustand zu zeichnen.

(52) **GDI** – Graphics Device Interface

(53) Die `ComboBox` ist eines der Beispiele, wo das Zeichnen vom Betriebssystem vollzogen wird. Hier ist es u.U. sinnvoller, solch ein Element direkt als User Control aus mehreren einzelnen Komponenten nachzubauen.

4.4 Softwarearchitektur

Dieser Abschnitt zeigt die **externe Sicht** auf das System **geolyzer**. Dabei handelt es sich um eine klassische **zweischichtige Client/Server-Architektur**. Da der SQL Server lediglich der Datenhaltung dient und die Anwendungslogik auf den Clients implementiert ist, handelt es sich um ein sog. **»Fat-Client-Modell«**. **Abbildung XII** verdeutlicht dies.

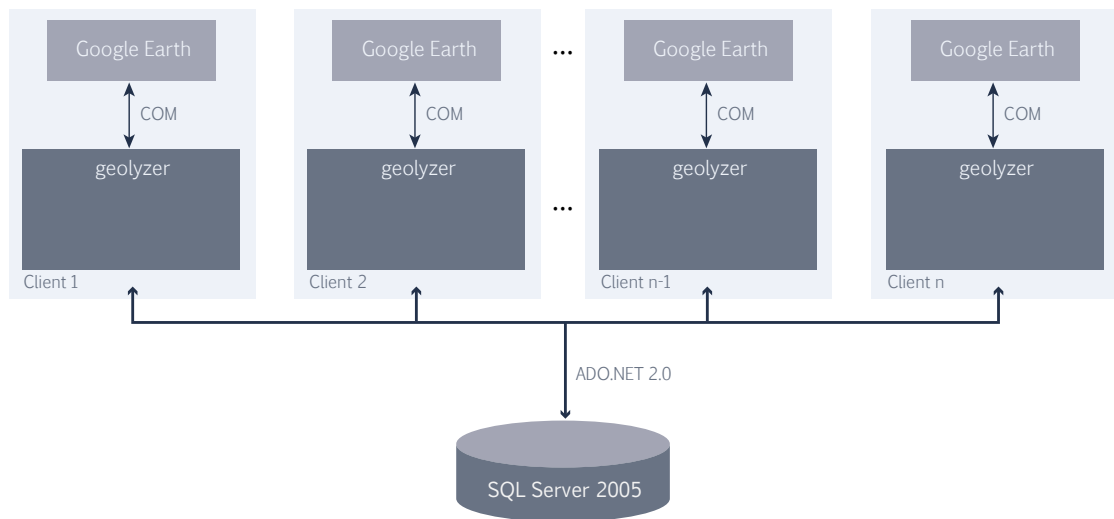


ABBILDUNG XII:
geolyzer – zweischichtige Architektur Client/Server

Um die Strukturierung des System in einzelne Komponenten widerzuspiegeln, zeigt **Abbildung XIII** das zugehörige Paketdiagramm des Ist-Zustandes.

- » **Interoperability** – enthält Pakete mit Klassen zur Zusammenarbeit verschiedener Systeme, d.h. zur Kommunikation der laufenden Anwendung mit Google Earth und dem Microsoft SQL Server 2005.
- » **GUI** – enthält Pakete der grafischen Benutzeroberfläche. Darunter fallen Benutzersteuerelemente (*intern entwickelt oder als Klassenbibliothek eingebunden*) sowie modale Dialogfenster (*Forms*).
- » **Application** – enthält Pakete mit Klassen der laufenden Anwendung: Bildbearbeitung, Threading, KML-Abbildung und Dokumenterstellung sowie Zugriff aufs Dateisystem.

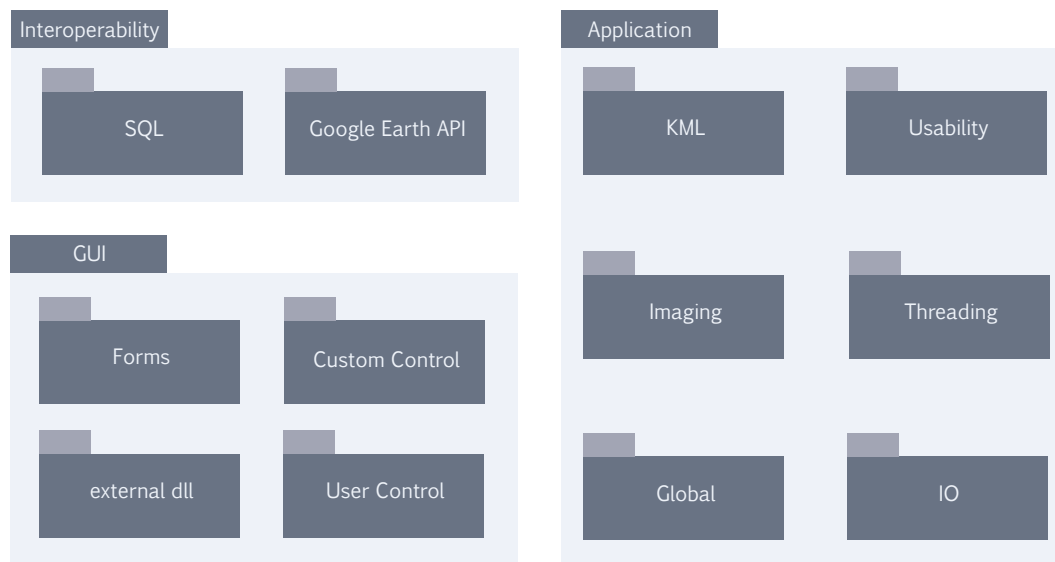


ABBILDUNG XIII:
geolyzer – Paketdiagramm

4.5 Soll-Zustand des Endprodukts

Nachdem der **Ist-Zustand** skizziert wurde, ist es an der Zeit, den **Soll-Zustand** für das Endprodukt zu definieren, um am Ende dieser Arbeit zu vergleichen, welche Funktionalität im Rahmen realisiert werden konnte und welche nicht.

Basierend auf den Ergebnissen der Usability-Verfahren wird unterschieden zwischen **quantitativen, Tracking** (*als Teilmenge der quantitativen Verfahren*) und **heuristischen Verfahren – qualitative Verfahren** lassen sich nicht umsetzen. Als neuer Typ werden **auto-evaluierende Verfahren** eingeführt, die das rechnerbasierte Bewerten der Usability bezeichnen. Der **Aufwand** ist geschätzt und basiert auf Erfahrungswerten.

Funktion	Beschreibung	Typ	Aufwand
<i>Feedback</i>	Statistische Auswertung und visuelle Darstellung des Nutzerfeedbacks in Diagrammform. Export von Statistiken als Report im Portable Document Format.	quantitativ	moderat
<i>Nutzeraktivität</i>	Visualisierung aller Nutzerereignisse wie Kamerafokus, Anwählen von Ortsmarken, Suchfunktionen etc. und Export als KML-Datei.	tracking	gering
<i>3D-Heatmap</i>	Quantitative Abbildung der Maus-Aktivität auf dem Globus während einer Aufnahme und Export als KML-Datei mit 3D-Objekten, die dieses wiedergeben.	tracking	hoch
<i>Validierung</i>	Validierung des KML-Markups nach Kriterien bzgl. der Zugänglichkeit und der Gebrauchstauglichkeit. Der Anwender soll ein Report mit Mängeln und Verbesserungsvorschlägen erhalten.	heuristisch	gering bis moderat
<i>Klassen</i>	Ab einer entsprechend großen Menge an Aufzeichnungen zu einer KML lassen sich Klassen bilden, die Unregelmäßigkeiten und Gemeinsamkeiten bei der Aufnahme repräsentieren. Diese lassen sich in Kombination mit den anderen Verfahren zur Bewertung des KML-Materials nutzen. Auch dies soll als Report exportiert werden können.	evaluierend	hoch

TABELLE IV: Soll-Funktionalität geolyzer

Eine genaue Beschreibung der einzelnen Funktionen erfolgt im nächsten Kapitel zur Konzeption und Realisierung des **geolyzer** Endprodukts.

“Knowledge can be communicated,
but not wisdom.”

Hermann Hesse
German Novelist

5. Entwurf und Realisierung

Was sich am Anfang einer Chronologie als großes Hindernis präsentiert, entpuppt sich später oft als Spaziergang. Und die Krux, die anfangs noch überschaubar scheint, entwickelt sich mit fortschreitender Zeit zu einem lähmenden Geflecht. Auch wenn sich anbahnende Ereignisse im Laufe eines Prozesses mit bereits erlernten Mustern divinatorisch erfassen lassen, so lehrt gerade die Erfahrung, dass man sich auf sie allein eben nicht verlassen sollte. »Expect the unexpected« – eine Universal-Empfehlung für ein ganzes Leben, und besonders für diese Arbeit, denn einiges kam anders als ursprünglich gedacht – im positiven wie im negativen Sinne.

Dieses Kapitel beschreibt Konzeption und Umsetzung der Usability-Komponenten. Um die Software geolyzer entsprechend zu testen, erfolgt neben den üblichen Softwaretests ein Testszenario mit Probanden in Zusammenarbeit mit dem bereits erwähnten Bremer Start-Up Nuterra und der Oldenburger visuamundo GbR. Nur so lässt sich anschließend belegen, ob der geolyzer den Praxistest besteht oder nicht – anders ausgedrückt, ob sich der Aufwand gelohnt hat und interessierten Unternehmen tatsächlich ein einfaches und aussagekräftiges Kompakt-Usability-Labor in die Hand gedrückt wird oder – im schlechten Fall – die erhaltenen Ergebnisse keine oder zu geringe Aussagekraft haben, so dass es nicht vertretbar wäre, für die Nutzung der Software irgendjemanden einen Cent in Rechnung zu stellen.

5.1 Konzept der Usability-Komponenten

Der Sinn dieser Arbeit besteht darin, durch Untersuchung gängiger Usability-Prinzipien und Verfahren ein passendes Konzept zur Untersuchung der Gebrauchstauglichkeit für Geobrowser-Inhalte zu entwickeln, welches in das System **geolyzer** integriert wird. Die Untersuchung der Usability-Prinzipien hat ergeben, dass mit Ausnahme der qualitativen Verfahren die anderen Verfahren sich für ein automatisches Werkzeug eignen. Die Abbildung auf Geobrowser-Inhalte soll im Folgenden geschehen. Neben den **quantitativen** und **heuristischen** Usability-Verfahren werden **auto-evaluierende Verfahren** als Novum eingeführt, die einen Versuch wagen, die Usability automatisch zu bewerten. Somit erscheint eine Differenzierung der Funktionalität in die Bereiche Quantitativ, Heuristisch und Auto-Evaluierend sinnvoll.

5.1.1 Konzept geolyzer – Quantitativ

Basierend auf den quantitativen Verfahren zur Untersuchung von Inhalten soll diese Komponente einerseits **Tracking** umsetzen und ebenso das **Fragebogen-Feedback-Prinzip** verwirklichen. Beim Tracking handelt es sich um die reine Aufzeichnung der Nutzeraktivität (*Nutzerdaten*) – es beinhaltet keine automatische Bewertung des aufgezeichneten Materials.

Nutzerdaten untergliedern sich beim System **geolyzer** in die folgenden Kategorien:

- » **Kamerafokus**
- » **Maus-Aktivität** auf dem Globus
- » **Nutzerereignisse** – Suche, Anwählen von Ortsmarken, Ebenen ein- und ausschalten

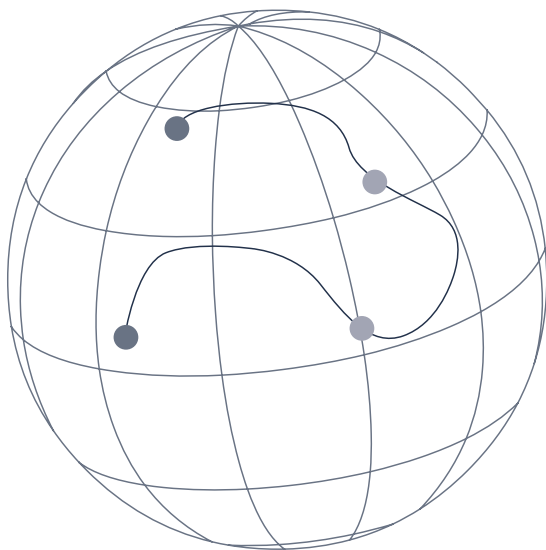
Die Aufzeichnung der Daten wird entsprechend visualisiert. Und hier bietet sich mit der KML eine geeignete Struktur an, die wenig Speicherplatz benötigt und an das jeweilige Umfeld – die gängigen Geobrowser – angepasst und somit kompatibel ist. Die erste Trackingfunktion besteht aus der **visuellen Aufbereitung** des **Kamerafokus** sowie den **Nutzerereignissen**, die zweite aus der Abbildung der **Maus-Aktivität** auf dem Globus als **dreidimensionale Heatmap**.

Die Aufzeichnung der Daten wurde bereits insoweit vor Beginn der Diplomarbeit realisiert, als dass das aktuelle Kameraobjekt sowie die Nutzerereignisse während einer Aufnahme in der Datenbank persistent gespeichert werden und der zeitliche Kamerafokus mithilfe des **geoPlayers** wiedergegeben werden kann.

Hier bleibt nur die Abbildung der Daten auf das KML-Format und die damit einhergehende Visualisierung zu realisieren, weshalb dieser Schritt bei der Einschätzung des Aufwands auch die Bewertung »gering« erhielt.

Tracking – Nutzeraktivität

Um wiederzugeben, welche Aktionen der Proband während einer Aufnahme durchgeführt hat, wird der Kamerafokus – das ist der Punkt auf dem Globus, auf den die Kamera gerichtet ist – durch **Vektordaten** auf der virtuellen Erdkugel visualisiert, mit einer **Start-** und **Endmarke** versehen und alle aufgetretenen **Ereignisse** werden mit Beschreibung und Zeitstempel auf dem Verlauf des Kamerafokus platziert. **Abbildung XIV** zeigt eine vereinfachte Skizze dieses Prinzips.



- Ereignis-Ortsmarken
- Start- bzw. Endpunkt

ABBILDUNG XIV:
Skizze zur Abbildung der Nutzeraktivität

Aus der Datenbank lässt sich der Kamerafokus zu einer Aufnahme als Tabelle aus zeitlich zusammenhängenden Koordinaten auslesen. Einer gespeicherten Funktion wird die ID der Aufnahme übergeben und es wird eine Tabelle folgender Struktur zurückgegeben:

Longitude	Latitude
1.42616008134189	48.72431282586730
1.42616008134189	48.72431282586730
1.78701007663743	49.00753931838160

Dabei entsprechen die Koordinaten im `DataTable` an der Position `[0]` und `[n]` der Start- bzw. Endkoordinate der Aufnahme. Mithilfe des KML `LineString`-Tags lässt sich aus den erhaltenen Koordinatenwerten ein zusammenhängender Vektorpfad auf dem Globus erzeugen. Hier ist Folgendes bei der Realisierung zu beachten:

- » Bei Aufnahmen gibt es zeitliche **Phasen ohne Kameraaktivität**. Um hier nicht unnötige Koordinaten – nämlich identische – im `LineString`-Tag zu platzieren, müssen Koordinaten **gefiltert** werden. Des Weiteren sollen nicht nur identische, sondern auch **sehr nah** aneinander **liegende Koordinaten verworfen** werden, was am Endergebnis optisch nichts ändert, aber die Dateigröße erheblich verkleinert. Für solch einen Grenzwert ist zwingend zu beachten, dass der Abstand⁽⁵⁴⁾ der Kamera zum Fokuspunkt auf dem Globus (*Focus Range*) mit in die Berechnung einfließen muss.

Es wird darauf hingewiesen, dass mit dem Vektorpfad bislang nur der **Fokuspunkt der Kamera**, nicht aber deren **Position** abgebildet wird. Die Google Earth COM API stellt leider keine Methode bereit, die die tatsächliche Position der Kamera zurückgibt. Im Sinne der Usability-Analyse ist es zwar wichtiger, den Fokuspunkt anstelle der Position darzustellen – andererseits könnte ein zusätzlicher Vektor, der vom Fokuspunkt zur Kameraposition gerichtet ist, das Ergebnis aufwerten und Analysten die Kameraposition offenbaren. Diese lässt sich durch trigonometrische Berechnungen mithilfe von `focusRange`, `tilt` sowie `azimuth` realisieren. Aus zeitlichen Gründen wird vorerst ein vereinfachtes Modell benutzt, welches die Position der Kamera direkt im Winkel von 90° mit dem Abstand `FocusRange` über dem Fokuspunkt abbildet.

(54) Eine Navigation sehr nah an der Erdoberfläche bedeutet einen kleineren Abstand als eine Navigation aus der Weltraumperspektive – sehr nah aneinander liegend ist somit relativ zum Abstand der Kamera zum Globus.

Da beim **geolyzer** eine Sitzung mehrere Aufnahmen beinhalten kann, soll beim Export der Nutzer entscheiden, ob nur die aktuell im **geoPlayer** angewählte Aufnahme oder alle Aufnahmen einer Sitzung in die exportierte KML-Datei mit einfließen.

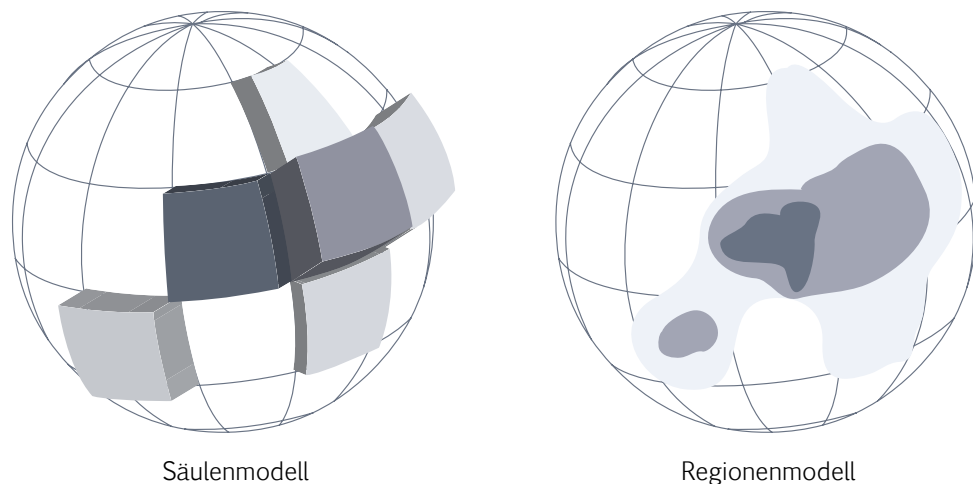
Tracking – 3D-Heatmap

Das 3D-Korrelat zu 2D-Heatmaps soll vom Prinzip her nichts anderes leisten. Die Google Earth COM API bietet die wichtige Funktionalität zum Abfragen der **Globuskoordinate** zur **normalisierten Bildschirmkoordinate**. Gäbe es diese Möglichkeit nicht, wäre es unmöglich eine Heatmap zu erzeugen, da nicht der Kamerafokus von Relevanz ist, sondern die projizierten Maus-Koordinaten auf dem Globus. Es werden somit während einer Aufnahme zusätzlich zu **Kamera-** und **Ereignisdaten** auch projizierte **Globuskoordinaten** zur Maus-Position in der Datenbank gespeichert. Für die visuelle Erzeugung einer 3D-Heatmap bieten sich mindestens zwei Möglichkeiten an.

- » Dort auf dem Globus, wo der Maus-Zeiger besonders oft verweilt, soll eine **3D-Säule** entstehen, wobei **Länge** und **Farbe** Aufschluss über Verweildaten liefern können. Wenig Aktivität führt zu einer Säule geringer Höhe und viel Aktivität resultiert in einer Säule mit deutlicher Höhe. Mit der Farbe kann eine weitere Information abgebildet werden.
- » Anstelle von Säulen werden **verschiedenfarbige Polygone** gezeichnet, welche die Regionen umfassen, in denen ähnlich viel Maus-Aktivität stattfand.

Die zweite Variante mit den Regionen würde mehr dem 2D-Korrelat entsprechen und hätte dieselbe Aussagekraft wie die erste Variante. Sie ist deutlich aufwändiger zu realisieren, da hier ein mathematisches Verteilungsmodell implementiert werden müsste, welches die gespeicherten Koordinaten zu Regionen ähnlicher Aktivität zusammenfasst. Des Weiteren würden sich die Regionen überlagern, würde man nicht Polygone mit inneren und äußeren Grenzen (OuterBoundary und InnerBoundary-Tags) definieren – dieses umzusetzen ist ebenfalls ein enormer Rechenaufwand. Letztlich führt diese Variante aufgrund der hohen Koordinatenzahl einzelner Polygone zu einem großen Overhead, der die Dateigröße schon bei mittelgroßen Heatmaps deutlich erhöht.

Die erste Variante ist schlanker und lässt sich schneller umsetzen. Hier ist ein **Rastermodell des Globus** vonnöten, auf dem sich die Maus-Aktivität gemerkt wird. Die Anzahl an horizontalen und vertikalen Unterteilungen wird so gewählt, dass der Rechenaufwand vertretbar bleibt und die Säulen ihre Aussagekraft behalten – was bei einem groben Raster wie in der unteren Skizze dargestellt nicht der Fall wäre. Da die Maus-Aktivität auf das Raster abgebildet wird, müssen nicht die Maus-Koordinaten in der Datenbank gespeichert werden, sondern nur die **jeweiligen Regionen** mit zwei **numerischen Werten**⁽⁵⁵⁾, die die Maus-Aktivität repräsentieren. **Abbildung XV** zeigt die beiden Konzepte als stark vereinfachte Skizze.



Säulenmodell

Regionenmodell

ABBILDUNG XV:
Zwei Konzepte zur Visualisierung von Heatmaps

Die Variante mit den **3D-Säulen** wird realisiert. Da es für eine 3D-Säulen-Heatmap zwei Möglichkeiten gibt, Informationen zu visualisieren – über die **Höhe** und über die **Farbe** – soll über die Höhe dargestellt werden, **wie oft** sich der Maus-Zeiger über einer Region befand und die Farbe zeigt an, **wie lange** der Mauszeiger über dieser Region durchschnittlich verweilte. Der Grund ist, dass es im Sinne der Usability-Analyse einen Unterschied macht, ob ein Maus-Zeiger eine Region besonders oft besucht und danach gleich wieder zu einer anderen Region navigiert oder ob der Maus-Zeiger an ein und derselben Region besonders lange verweilt.

(55) Ein Wert repräsentiert die Häufigkeit, ein anderer die Verweildauer des Mauszeigers.

Wie beim Export des Verlaufs des Kamerafokus entscheidet der Nutzer, ob nur die aktuell im **geoPlayer** angewählte Aufnahme oder die gesamte Sitzung exportiert werden soll.

Quantitativ – Feedback

Bereits jetzt bietet die Software die Möglichkeit, für jede Sitzung ein individuelles **Nutzerfeedback** zu definieren, d.h. es wird ein **Fragenkatalog** mit den jeweiligen Antworttypen erstellt, der Probanden im Anschluss einer Aufnahme zur Beantwortung übergeben wird. Der Antworttyp ist untergliedert in folgende Typen:

- » **Bewertung** – Auswahl der Antwortmöglichkeiten »*sehr gut*« bis »*sehr schlecht*«
- » **Freitext** – Freier Text als Antwort
- » **Auswahl** – Auswahl aus von Analysten vordefinierten Antwortmöglichkeiten

Der Antworttyp Bewertung lässt sich gut als **Tortendiagramm** statistisch aufbereiten. So sind Analysten in der Lage, für jeden Probanden einen schnellen Überblick zu erhalten, wie Probanden durchschnittlich bewerten. Das ist für Usability-Tests insofern wichtig, dass auf Grundlage dieser Statistik vor einem geplanten Test eine gemischte Probandengruppe angelegt werden kann – sofern genügend Nutzerdaten in der Datenbank vorhanden sind.

Es wird in der Komponente zur Probandenverwaltung ein Benutzersteuerelement »*Diagramm*« mithilfe von **GDI+** entwickelt, welches die **durchschnittliche Bewertung** visualisiert. Neben diesem Steuerelement werden in einem `ListView`-Steuerelement **Statistiken**⁽⁵⁶⁾ zu Nutzern darstellt und in einem dritten `RichText`-Steuerelement wird jegliches **Feedback** als Übersicht zu einem Probanden aufgelistet.

Des Weiteren muss das Programm die Funktionalität bieten, all diese genannten Daten visuell aufbereitet als **Report** im **Portable Document Format** (PDF) zu exportieren. Der Nutzer kann diesen Report einmal direkt aus der Probandenverwaltung für den jeweiligen Probanden erzeugen, des Weiteren aus der Menü-Leiste über ein zusätzliches Dialogfeld, in welchem angegeben wird, ob entweder alle oder eine selbst definierte Menge an Probanden dem Report hinzugefügt werden.

(56) Statistiken wie Anzahl an Aufnahmen etc. sind für die Usability und somit im Rahmen dieser Arbeit nicht relevant und werden auch nicht weiter erwähnt. Sie werden dennoch parallel zur Arbeit realisiert, da diese Informationen für Analysten wichtig sein können.

5.1.2 Konzept geolyzer – Heuristisch

Im Sinne der Heuristik soll diese Komponente Inhalte auf **Einhaltung** von **Konventionen** überprüfen. Dazu zählen hauptsächlich die **Usability-** und **Accessibility-Richtlinien** aus dem Web-Bereich, die sich auch auf Geobrowser-Inhalte übertragen lassen. Für diese Komponente wird eine geeignete Teilmenge der »*Usability-Guidelines*« der US-Regierung [GUIDELINES 2006] sowie einige Richtlinien von **Jakob Nielsen** und **Vitaly Friedman** übernommen. Hinzu kommen Konventionen für 3D-Geobrowser-Inhalte, die eigens zu diesem Zweck definiert wurden. Diese basieren auf den Erkenntnissen des Kapitels »*Geobrowser und Usability*«.

Die folgende Tabelle beschreibt die infrage kommenden Heuristiken mit der Art der **Validierung** sowie dem zugrunde liegenden **Typ**. Der Typ bestimmt, ob es sich um Heuristiken aus dem Web-, dem Accessibility-Bereich oder um Konventionen für 3D-Geobrowser handelt. Bei der Validierung wird unterschieden zwischen **automatischer Validierung** und der **Validierung** durch **Probanden**. Gewisse Kriterien lassen sich nicht automatisch überprüfen, so dass hier die Meinung eines Probanden vonnöten ist.

Beschreibung	Validierung	Typ	Umsetzung
Da KML auf XML basiert, soll das Markup einer KML-Datei auf Wohlgeformtheit überprüft werden.	automatisch	3D-Geobrowser	Indirekt umgesetzt, da geolyzer beim KML-Import die Wohlgeformtheit überprüft
Placemark-Tags sollten über das LookAt-Tag verfügen	automatisch	3D-Geobrowser	Klasse Usability_Heuristic_Auto
Inhalte durch Folder-Tags ab einer gewissen Menge an Placemarks strukturieren	automatisch	3D-Geobrowser	Vorerst nicht umgesetzt
Kontrastreiche Text- und Hintergrundfarben	automatisch	Accessibility	Klasse Usability_Heuristic_Auto
Nicht-Text Elemente sollten über alternativen Text verfügen	automatisch	Accessibility	Vorerst nicht umgesetzt

Beschreibung	Validierung	Typ	Umsetzung
Abkürzungen im Text durch <code><abbr></code> -Tags kennzeichnen	automatisch	Accessibility	Vorerst nicht umgesetzt
Tastaturkürzel für Navigationselemente mithilfe von <code>accesskey</code> bestimmen	automatisch	Accessibility	Vorerst nicht umgesetzt
Schriftgrad durch <code></code> -Angaben an Nutzerbedürfnisse anpassbar machen	automatisch	Accessibility	Vorerst nicht umgesetzt
Navigationsreihenfolge für Tabulatortaste durch <code>tabIndex</code> ermöglichen	automatisch	Accessibility	Vorerst nicht umgesetzt
Farbe nicht als alleiniger Indikator zur Visualisierung wichtiger Inhalte	Proband	Accessibility	Klasse <code>Usability_Heuristic_User</code>
Start-Ortsmarke soll Sinn und Absicht der Geobrowser-Präsenz dem Nutzer klar und unmissverständlich mitteilen	Proband	Accessibility 3D-Geobrowser	Klasse <code>Usability_Heuristic_User</code>
Jede Ortsmarke sollte einen Link zur Start-Ortsmarke realisieren und Nutzern mitteilen, wo sie sich gerade befinden	Proband	Accessibility Web-Heuristik 3D-Geobrowser	Klasse <code>Usability_Heuristic_User</code>
Die visuelle Konsistenz der einzelnen Inhalte sollte sichergestellt sein. Das gilt insbesondere für Icons	Proband	Accessibility Web-Heuristik 3D-Geobrowser	Klasse <code>Usability_Heuristic_User</code>
Verschiedene Inhalte sollten sich nicht überdecken	Proband	Accessibility 3D-Geobrowser	Klasse <code>Usability_Heuristic_User</code>
Die Sprache des Benutzers soll verwendet werden	Proband	Accessibility	Klasse <code>Usability_Heuristic_User</code>
Gewährleistung der Wiedererkennung von Seitenelementen	Proband	Web-Heuristik	Klasse <code>Usability_Heuristic_User</code>

TABELLE V: Konzept **geolyzer** Heuristisch

Um **Heuristiken** auszuwerten, die **nicht automatisch** überprüfbar sind, wird die Meinung der Probanden *»abgeholt«*. Die Lösung liegt mit dem **benutzerdefinierten Fragenkatalog** der quantitativen Verfahren nahe. Es wird ein zusätzlicher Typ *»heuristic«*⁽⁵⁷⁾ erschaffen, der gezielt heuristische Fragen beinhaltet, die als Antwortmöglichkeit den **Grad der Zustimmung** verlangen – dabei handelt es sich nicht um eine Bewertung von gut bis schlecht, sondern um die Bandbreite *»volle Zustimmung«* bis *»keine Zustimmung«*.

Diese Komponente wird modular aufgebaut, so dass sich in späteren Versionen beliebig viele Heuristiken – automatischer oder manueller Natur – einfach ins Programm integrieren lassen. Gemäß der Tabelle werden im Rahmen dieser Arbeit nur die **zwei wichtigsten** automatisch zu validierenden Heuristiken realisiert. Dazu zählt die Validierung des **Farbkontrastes** sowie die Untersuchung, ob Ortsmarken über LOOKAT-Tags verfügen.

Bei **Accessibility-Heuristiken**, wie z.B. Tags zu Abkürzungen oder alternativen Text, besteht derzeit noch Unklarheit, inwieweit assistive Technologien überhaupt in der Lage wären, barrierefreies Markup im Geobrowser zu interpretieren. Gemäß dem GoogleWatch-Blog⁽⁵⁸⁾ wird Barrierefreiheit in Google Earth noch zu wenig unterstützt. Laut **T.V. Raman**, einem blinden Mitarbeiter von Google, wird sich das eher früher als später ändern, da auch die meisten anderen etablierten Google Produkte barrierefrei sind. Aus diesem Grund, aber auch aus zeitlichen Gründen, werden diese Heuristiken, die sich fast analog zu den anderen Heuristiken umsetzen lassen, vorerst nicht realisiert.

Die **3D-Geobrowser-Heuristik** zur Validierung, ob Inhalte ab einer gewissen Menge an Ortsmarken durch Verzeichnisse strukturiert sind, wäre vom Aufwand der Realisierung noch vertretbar, wird aber vorerst nicht umgesetzt, da der Bedarf an dieser Heuristik⁽⁵⁹⁾ zu gering ist.

Der Anwender hat beim Export als PDF-Datei die Möglichkeit, aus einer Liste alle gewünschten Heuristiken auszuwählen, die fortan im Report aufgeführt werden. Die Heuristiken vom Typ Proband werden der Einfachheit halber zum Typ **Probanden-Heuristik** zusammengefasst.

(57) Da diese Fragen allgemeiner Natur sind und sich für jegliche Inhalte nutzen lassen, werden sie fest in das Programm integriert – d.h. sie müssen nicht vom Analysten gesondert angelegt werden.

(58) siehe dazu <http://www.googlewatchblog.de/tag/barrierefrei>

(59) Laut nuterra und der visuamundo GbR, die Inhalte in Geobrowsern aufbereiten, soll die Navigation in Zukunft nur noch in den Ortsmarken erfolgen, so dass kein wirklicher Bedarf mehr an Verzeichnissen besteht.

5.1.3 Konzept geolyzer – Auto-Evaluierend

Mit den ersten beiden Konzepten wurden bekannte Usability-Verfahren an Geobrowser-Inhalte angepasst. Mit der Visualisierung der Mausaktivität durch 3D-Objekte wurde Tracking in die dritte Dimension überführt, beruht dabei aber weiterhin auf dem Prinzip der Aufzeichnung und Wiedergabe von Nutzerereignissen.

Mit dem letzten Konzept dieser Arbeit, als **Auto-Evaluierend** bezeichnet, wird Neues gewagt. Es gilt, dass sich der Grad an Barrierefreiheit bei Web-Inhalten durch die Überprüfung des Markups automatisch evaluieren lässt. Für die Usability, so sind sich Experten einig, ist eine auf automatischem Wege erhaltene qualitative Bewertung nicht möglich, da hier ein Probandentest mit anschließender »manueller« Auswertung des Verhaltens, der Empfindungen sowie der Meinungen der Probanden unabdinglich ist. Auch die automatische Auswertung von Heuristiken ergab, dass eine Evaluation *ohne* Proband nur dort möglich ist, wo es direkt oder indirekt um das Markup von KML-Inhalten geht.

Um trotzdem den Anwendern von **geolyzer** als Ergebnis einen Report zur Verfügung zu stellen, der im Gegensatz zum bislang Genannten nicht nur eine Auflistung und Wiedergabe von Merkmalen enthält, sondern insbesondere eine Bewertung des KML-Materials, so muss wie bei einem manuell durchgeführten Test das Verhalten sowie die Meinung der Probanden in die Untersuchung mit einbezogen werden.

Bevor im Folgenden die zugrunde liegende Idee der Auto-Evaluation näher erläutert wird, soll zuerst definiert werden, wie die erhaltenen Ergebnisse aufbereitet werden sollen. Hier kommt das in Kapitel 3.4 eingeführte **CIF** zum Einsatz, d.h. der Report wird nach Art des **Common Industry Formats** automatisch erstellt. Die folgende **Tabelle VI** stellt ausgewählte **Reportdaten**⁽⁶⁰⁾ des CIF für den Zweck dieser Komponente dar und beschreibt, wie **geolyzer** an diese Daten gelangt – d.h. ob bereits vorhandene Funktionalität genutzt werden kann, wie es beispielsweise bei einigen gespeicherten Funktionen der Fall ist, oder zusätzliche Funktionalität erst noch erschaffen werden muss.

(60) Nicht alle Reportdaten, die vom NIST für das CIF definiert wurden, werden im Report enthalten sein. Einige beziehen sich nur auf Web-Inhalte, andere haben im Sinne der Usability-Evaluation keine Aussagekraft – z.B. ob Probanden bezahlt und über Rechte aufgeklärt wurden.

Abschnitt	Reportdaten	Funktionalität vorhanden?
Deckblatt	Produktname	Nein. Sitzungsdaten erweitern
	Testverantwortliche Person	Nein. Neues Dialogfeld Reportexport
	Datum der Durchführung	Ja. Daten der vorhandenen Aufnahmen
	Name des Kunden	Nein. Sitzungsdaten erweitern
	Kontaktdaten für Kunden	Nein. Neues Dialogfeld Reportexport
Übersicht	Testbeschreibung	Ja. Beschreibung einer Sitzung
	Statistisches (u.a. Probandenzahl)	Ja. Gespeicherte Funktionen vorhanden
Produktbeschreibung	Produktname	Nein. Sitzungsdaten erweitern, s.o.
	Zielgruppe	Nein. Sitzungsdaten erweitern
	Einsatzumgebung	Nein. Sitzungsdaten erweitern
Testbeschreibung	Auflistung aller Aufgaben	Ja. Gespeicherte Funktionen vorhanden
	Testort	Nein. Zusätzliche Eingabemaske für Aufnahmen
	Statistisches	Ja. Gespeicherte Funktionen vorhanden, s.o.
	Nutzergruppen	Ja. Nutzerprofil
Metriken	Verwendbarkeit	Nein. Konzept Auto-Evaluation
	Effektivität	Nein. Konzept Auto-Evaluation
	Zufriedenheit	Nein. Konzept Auto-Evaluation

TABELLE VI: Report auto-evaluierend gemäß CIF

Um die fehlenden **CIF-Metadaten** zu Usability-Tests aufnehmen zu können, sind die in der Tabelle genannten Anpassungen und Erweiterungen des Programms zu vollziehen, die parallel umgesetzt und im Folgenden nicht weiter erwähnt werden.

Weitaus interessanter sind die **Metriken**, die sich in die Kategorien **Verwendbarkeit**, **Effektivität** und **Zufriedenheit** untergliedern. Das CIF definiert diese wie folgt:

Verwendbarkeit

- » Prozentualer Anteil an erfolgreich erfüllten **Aufgaben**⁽⁶¹⁾ (*Completion Rate*)
- » Häufigkeit von **Fehlern** (*Error Rate*)
- » Anzahl an **Hilfestellungen** während eines Tests (*Assist Rate*)

Effizienz

- » Zeit, die für einzelne Aufgaben benötigt wird (*Mean Task Time*)
- » Erfüllte Aufgaben / Durchschnittszeit (*Efficiency Product Level*)

Zufriedenheit

- » Durchschnittliche **Zufriedenheit** der Probanden (*Satisfaction Rate*)

Von diesen Metriken lässt sich der prozentuale Anteil an **erfolgreich erfüllten Aufgaben** sowie die Metrik zur **Zufriedenheit** der Probanden realisieren. **Fehler** und **Hilfestellungen** während eines Tests werden von **geolyzer** aktuell nicht unterstützt. Ebenso stellt das Programm keine Möglichkeit zur Verfügung, die **Zeit** für einzelne Aufgaben zu messen. Nichtsdestotrotz wäre es möglich, diese Funktionalität in das Programm zu integrieren. Im Rahmen dieser Arbeit wird dies aus zeitlichen Gründen nicht geschehen, da der Eingriff in das Programm und der damit verbundene Aufwand das Ergebnis nicht rechtfertigen.

Da mit diesem Konzept hauptsächlich etwas Neues gewagt wird, und es auch nicht herausfordernd wäre, nur die CIF-Metriken umzusetzen, sollen gerade hier neue Ansätze erdacht und erprobt werden. Da die Hauptfunktion von **geolyzer** bekanntlich aus der Aufzeichnung der Nutzeraktivität besteht, soll auf dieser Grundlage das vorhandene Material **automatisch interpretiert** werden.

(61) **Wichtig:** Gestellte Aufgaben sollen sich auf Inhalte und nicht auf Rahmenbedingungen beziehen.

Interpretation Kamerafokus

Durch den Verlauf des Kamerafokus sind Daten vorhanden, die wiedergeben, **welche Inhalte** Probanden während einer Aufnahme **fokussiert** haben. Hier gilt es zu vergleichen, sofern der Begriff »Inhalte« vorerst auf Ortsmarken beschränkt wird, welche dieser Marken innerhalb des Fokus lagen und welche nicht – anders ausgedrückt, welche Marken wurden vom Probanden beachtet und welche wurden vernachlässigt.

Für eine **Aufnahme** ergibt dies ein exemplarisches **Einzelergebnis** folgender Art:

- » **Vorhandene Ortsmarken:** 42
- » **Vom Probanden betrachtete Ortsmarken:** 13 (31%)

Für eine **Sitzung** werden aus den Daten der einzelnen Aufnahmen **Klassen** gebildet:

- » **Klasse D:** Ortsmarken mit insges. 0–24% Fokus: [Auflistung Ortsmarken]
- » **Klasse C:** Ortsmarken mit insges. 25–49% Fokus: [Auflistung Ortsmarken]
- » **Klasse B:** Ortsmarken mit insges. 50–74% Fokus: [Auflistung Ortsmarken]
- » **Klasse A:** Ortsmarken mit insges. 75–100% Fokus: [Auflistung Ortsmarken]

Die hiermit definierte Einteilung der Resultate in **vier Klassen** von **A** bis **D** soll für diese Prototypkomponente vorerst ausreichen. **Klasse A** gilt als positiv, **D** dementsprechend als eher negativ – wenn Ortsmarken der Klasse D nicht beachtet werden, kann dies mehrere Gründe haben und sagt vorerst nichts über die Qualität der Marke aus. Für einen Analysten ist diese Klassifikation ein Hinweis, **gewisse Inhalte genauer zu betrachten** und die Gründe herauszufinden – z.B. ob die Navigation mangelhaft ist und Inhalte deshalb erst gar nicht erreicht werden konnten. Um mehr Aussagekraft zu erhalten und zu den erstrebten Metriken zu gelangen, ist im Folgenden die **zeitliche Komponente** hinzuzuziehen.

Interpretation Heatmap

Durch die Aufzeichnung der Verweildauer des Mauszeigers sind Daten vorhanden, die bestimmen, welche Inhalte von Probanden **wie lange insgesamt** betrachtet wurden. Generell gilt, dass eine **lange Verweildauer** im Bereich einer Ortsmarke nicht zwangsläufig auf unverständliche Inhalte deutet – allein schon deshalb, da Ortsmarken sich inhaltlich quantitativ unterscheiden.

Durch die Heatmap sind Daten zu **Regionen** auf dem Globus vorhanden, die sich mit den **Positionen der Ortsmarken** abgleichen lassen. Zu beachten ist, dass die Region einer einzelnen Heatmap-Kachel allein nicht ausreichend ist, um die betreffende Ortsmarke zu beurteilen. Ortsmarken benötigen eine **endliche Menge** an **Heatmap-Kacheln** der **unmittelbaren Umgebung**, die dieser Ortsmarke zugeordnet werden.

Definiert man diese Menge als **aktive Region** einer Ortsmarke, so lässt sich durch die **Summe** an Hits – d.h. wie oft ein Mauszeiger über einer Kachel in einem festgelegten Zeitintervall registriert wurde – über den einzelnen Kacheln die **Gesamtzeit** ermitteln, die ein Proband bei dieser Ortsmarke verweilt hat.

$$\text{timeInterval} \cdot \sum_{i=1}^n \text{hTile_hit}_i$$

Weiterhin geben Heatmap-Daten Auskunft über die **höchste zusammenhängende Verweildauer**. Da Kacheln auf einen relativ kleinen Bereich beschränkt sind, lässt sich ab einer zu definierenden Verweildauer α behaupten, dass der Mauszeiger hier für n Sekunden geruht hat. Eine häufige **Mausinaktivität** muss aber noch nicht zwangsläufig darauf hindeuten, dass der jeweilige Proband Probleme mit dem Inhalt oder der Navigation hatte. Es kann ebenso sein, dass dieses Verhalten eine normale Navigationseigenschaft des betreffenden Probanden ist. Werden nun alle Ortsmarken der Aufnahme betrachtet, so lässt sich z.B. über die **Standardabweichung** σ erkennen, ob eine Ortsmarke mit **ungewöhnlich hoher Mausinaktivität** eine **Ausnahme** oder die **Regel** stellt. Anders ausgedrückt, es kann über das Navigationsverhalten des Probanden eine Aussage gemacht werden, ob bestimmte Ortsmarken vom Erwartungsverhalten abweichen.

Für eine **Aufnahme** hat ein exemplarisches **Einzelergebnis** folgende Struktur:

- » **Vorhandene Ortsmarken:** 42
- » **Höchste Gesamtzeit:** 63 Sekunden
- » **Geringste Gesamtzeit:** 3 Sekunden
- » **Median Gesamtzeit:** 14 Sekunden
- » **Höchste Inaktivität:** 9 Sekunden
- » **Geringste Inaktivität:** 0 Sekunden
- » **Median Inaktivität:** 0 Sekunden

Bei diesen Daten ist als **Lageparameter** das **arithmetische Mittel** \bar{x} wenig geeignet, da dieser Parameter als Vergleichswert dienen soll. Hat eine Aufnahme z.B. einmal eine höchste Inaktivität von 96 Sekunden und sieben Inaktivitäten von jeweils 0 Sekunden, so bietet sich der **Median** mit 0 Sekunden als Vergleichswert viel eher an als der Durchschnitt von $\bar{x} = 12.0$ Sekunden.

Der **Median** der **Inaktivität** aller Ortsmarken einer Aufnahme bestimmt, wie Nutzer erwartungskonform navigieren, d.h. ob sie bei der Navigation eher motorisch veranlagt sind (*geringer Wert*) oder sie einem Typus angehören, wo ruhende Mauszeiger beim Betrachten von Inhalten die Regel stellen (*hoher Wert*).

So kann für eine **Aufnahme** bereits für eine einzelne Ortsmarke bestimmt werden, ob sie **signifikant** vom Erwartungswert abweicht – was auf Unregelmäßigkeiten hindeutet.

Für eine **Sitzung** werden Ortsmarken in **Klassen** eingeteilt:

- » **Klasse D:** insgesamt sehr hohe Abweichung⁽⁶²⁾ [Auflistung Ortsmarken]
- » **Klasse C:** insgesamt überdurchschnittliche Abweichung [Auflistung Ortsmarken]
- » **Klasse B:** insgesamt normale Abweichung [Auflistung Ortsmarken]
- » **Klasse A:** insgesamt sehr geringe Abweichung [Auflistung Ortsmarken]

Eine Klassifizierung der Ortsmarken bezüglich der **Abweichung** von der **Gesamtzeit** ist wünschenswert. Hier kann allerdings nicht durch den Vergleich mit dem Median der Gesamtzeit bereits **innerhalb** einer **Aufnahme** eine Aussage getroffen werden, ob es sich um eine Irregularität handelt oder nicht – da Ortsmarken sich vom inhaltlichen Umfang her unterscheiden. Hier muss ein Abgleich mit den Daten einer **gesamten Sitzung** erfolgen.

Für eine **Sitzung** werden Ortsmarken auch hier in **Klassen** eingeteilt:

- » **Klasse D:** insgesamt sehr hohe Abweichung [Auflistung Ortsmarken]
- » **Klasse C:** insgesamt überdurchschnittliche Abweichung [Auflistung Ortsmarken]
- » **Klasse B:** insgesamt normale Abweichung [Auflistung Ortsmarken]
- » **Klasse A:** insgesamt sehr geringe Abweichung [Auflistung Ortsmarken]

(62) Eine Definition, welche Werte als »sehr gering« oder »überdurchschnittlich hoch« betrachtet werden, erfolgt im Kapitel zur Realisierung.

Bei der **Gesamtzeit** wäre es sinnvoll, wenn Analysten für zu untersuchende Ortsmarken bereits **im Vorfeld** eine Klassifizierung vornehmen könnten, welche Marken mehr und welche eher weniger Aufmerksamkeit vom Probanden erhalten sollen. So könnte durch die Abweichung verglichen werden, ob das Ergebnis im Erwartungsbereich liegt oder nicht. Der Eingriff in das Programm und der damit verbundene Aufwand ist sehr hoch, so dass dieses im Rahmen der Arbeit nicht umgesetzt werden kann.

Durch die **Klassifizierung** der Inhalte ist eine **Metrik** vorhanden, die darüber Auskunft gibt, ob das Nutzerverhalten einzelner Inhalte deutlich von dem der anderen Inhalte abweicht. Hier gilt, dass je mehr Aufnahmen eine Sitzung beinhaltet, umso höher die Aussagekraft dieser Metrik wird. Aus diesem Grund sollte der Umfang an Aufnahmen, den eine zu bewertende Sitzung umfasst, mindestens fünf⁽⁶³⁾ betragen. Bei einer geringeren Anzahl an Aufnahmen wäre die Aussagekraft dieser Metrik nicht gegeben.

Aus den genannten Metriken soll ein so genannter **geoIndex** von 0 bis 100 Punkten gebildet werden, so dass der Report neben den genannten Daten auch ein **numerisches Gesamtergebnis** in **Kurzform** enthält. Dieser Index beruht auf den einzelnen Ergebnissen, die in folgender **Tabelle VII** noch einmal übersichtlich dargestellt werden. Die **Gewichtung** ist vorerst abgeschätzt, da hier keinerlei Erfahrung vorliegt. A posteriori sollten diese Werte nach einer ausführlichen Praxisphase des **geolyzer** später kalibriert werden.

Metrik	Kurzbeschreibung	Anteil am Index
Completion Rate	Anteil an erfolgreich abgeschlossenen Aufgaben	15%
Satisfaction Rate	Probandenzufriedenheit. Wird durch Nutzer-Heuristiken ermittelt	15%
Heuristiken	Automatisch ermittelte Heuristiken, z.B. die Validierung des Farbkontrastes	10%
Fokus	Anteil an vom Nutzer fokussierten Ortsmarken	15%
Inaktivität	Inhalte mit signifikant abweichender Mausaktivität	30%
Gesamtzeit	Inhalte mit signifikant abweichender Gesamtzeit	15%

TABELLE VII: Index als Bewertung der Inhalte

(63) Laut Steve Krug sollte ein Usability-Test mindestens drei, besser allerdings fünf Testpersonen umfassen.

5.2 Implementierung und Integration

Nachdem das Konzept erstellt ist, gilt es nun, dieses mit den entsprechenden Werkzeugen zu realisieren und in den Ist-Zustand des **geolyzer** zu integrieren.

5.2.1 Realisierung geolyzer – Quantitativ

Da Tracking stets als KML-Datei exportiert wird, gilt es zuerst, eine geeignete Klasse zu implementieren, die in der Lage ist, für verschiedene Zwecke die **statische XML-Struktur** zu erzeugen und die **dynamische Struktur**, d.h. die Placemark- und Style-Tags, im Konstruktor als `List<KML_Placemark>` bzw. als `KML_Style` übergeben zu bekommen, um so ein **KML-Dokument** zu erzeugen. Diese Klasse soll aus Gründen der Performance die `IDisposable`-Schnittstelle⁽⁶⁴⁾ implementieren. **Abbildung XVI** zeigt die Klassenstruktur.

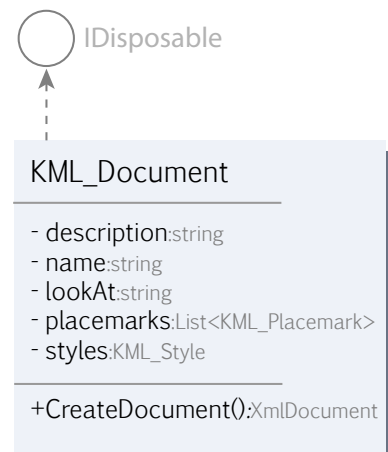


ABBILDUNG XVI:
Klasse `KML_Document`

Tracking – Nutzeraktivität

Im Paket **Usability** wird die Klasse `Usability_UserActivity_Export` implementiert, die die **Kameraobjekte** und **Ereignisse** vom SQL Server erhält und diese über die Klasse `KML_Document` als KML-Datei exportiert. Gemäß dem vorhandenen Konzept gilt, dass gleiche und sehr nah aneinander liegende Koordinaten zu **filtern** sind.

Ist gemäß **Abbildung XVII** der Winkel α kleiner als ein definierter Grenzwert, so wird $K(n+1)$ so lange verworfen bis α größer oder gleich dem Grenzwert ist. $K(n)$ und $K(n+1)$ sind die jeweils zu untersuchenden Kamerakoordinaten auf dem Globus. $K_{\text{Focus}}(n)$ ist dieselbe Koordinate wie $K(n)$, beinhaltet aber zusätzlich den Abstand (*FocusRange*) zum Globus. Um die Berechnung zu vereinfachen wird angenommen, dass sich die Position der Kamera im Winkel von 90° über dem Fokuspunkt befindet.

(64) Durch die Implementierung der `IDisposable`-Schnittstelle ist es möglich, Objekte innerhalb des using-Blocks zu erzeugen, so dass diese nach Beendigung des Blocks sofort aus dem Speicher entfernt werden – ohne auf die Aktivität des Garbage Collectors warten zu müssen.

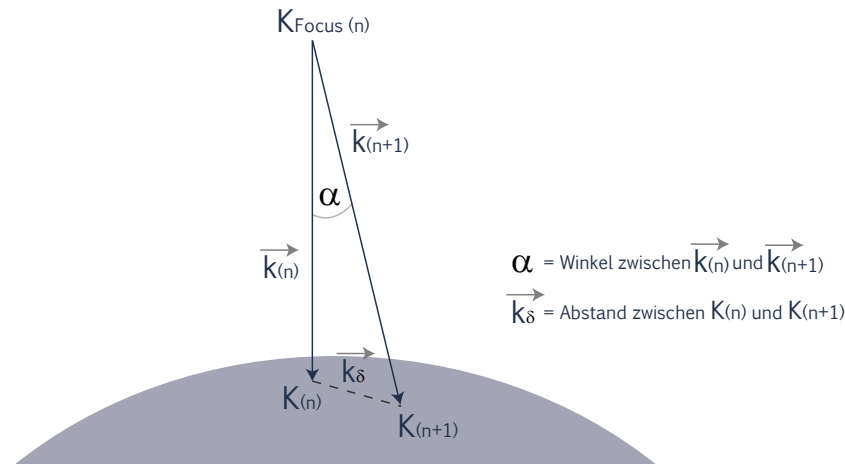


ABBILDUNG XVII:
Nah aneinander liegende Koordinaten filtern

Um dies zu realisieren sind Methoden vonnöten, die den Abstand k_{δ} zweier GPS-Koordinaten ermitteln und über die Umkehrfunktion des Tangens (*Arc Tangens*) den Winkel α berechnen.

Berechnung des Abstands (Km) zweier GPS-Koordinaten $K(n)$ und $K(n+1)$

```
public double GetGPSDistance(KML_GPS pos1, KML_GPS pos2) {}
```

Der Winkel Alpha zwischen $k(n)$ und $k(n+1)$

```
public double GetAlphaForAB(double distance, double focusRange) {}
```

Der Berechnung zufolge gelangen nur noch solche Koordinaten in die KML-Datei, wo der relative Abstand⁽⁶⁵⁾ zweier Koordinaten einen Grenzwert überschreitet – definiert durch den in mehreren Tests als geeignet zugewiesenem Grenzwert $\text{MIN_ANGLE} = 0.001^\circ$, so dass der gemessene Winkel α mit dem Grenzwert MIN_ANGLE verglichen wird. Die Höhe der Kamera über dem Boden $\vec{K}(n)$ wird in der Methode `GetAlphaForAB()` als `focusRange`, der Abstand zwischen $K(n)$ und $K(n+1)$ als `distance` übergeben.

(65) Zur Wiederholung: relativ bezieht sich auf den Abstand der Kameraposition zum Globus.

Tracking – Heatmap

Die Realisierung der 3D-Heatmap-Funktionalität beinhaltet die Lösung folgender Probleme:

- » Die **normalisierten Bildschirmkoordinaten** des Mauszeigers über dem `Panel`, in dem Google Earth eingebunden ist, werden benötigt. Aus diesen müssen **Globuskoordinaten** erzeugt werden.
- » Der **Globus** muss in ein **Raster** unterteilt werden, so dass erhaltene Globuskoordinaten einer Kachel auf dem Raster zugeordnet werden.
- » Die erhaltenen **Rasterdaten** müssen auf **Säulen** und **Farben** abgebildet und als **KML-Datei** exportiert werden.

1. Problem

Die Mauskoordinaten über dem Google Earth `Panel` werden benötigt: von $(0, 0)$ in der linken oberen Ecke bis $(\text{Panel.width}, \text{Panel.height})$ in der rechten unteren Ecke. Da das `Handle`⁽⁶⁶⁾ des `Panel`s durch die extern eingebundene Win32-Methode `SetParent()` im Besitz der Google Earth COM API liegt, gibt es keine Möglichkeit, Ereignisse wie z.B. `OnMouseOver` über dem `Panel` abzufangen und so an die Maus-Koordinaten zu gelangen.

Lösung

Es bleibt nur die Möglichkeit, über die statische `MousePosition`-Klasse die aktuellen Maus-Koordinaten abzufragen. Diese müssen auf Größe und Lage des `Panel`s projiziert werden. Dies muss nicht extra implementiert werden, da von `Windows.Form` abgeleitete Klassen mit der `PointToClient`-Methode eine Lösung bereitstellen.

```
Size s = this.panelGE.Size;
Point P = this.panelGE.PointToClient(new Point(MousePosition.X, MousePosition.Y));
```

Diese Koordinaten müssen durch folgende Umrechnung **normalisiert** werden, d.h. zu $(-1, 1)$ in der linken oberen Ecke, $(0, 0)$ im Mittelpunkt und $(1, -1)$ in der rechten unteren Ecke.

```
double dx = (P.X / (s.Width * 0.5f) - 1.0d);
double dy = 1.0d - P.Y / (s.Height * 0.5d);
```

(66) Ein `Handle` stellt einen Identifikator für digitale Objekte dar, beispielsweise einen Zeiger auf eine Adresse im Hauptspeicher.

Um die zugehörige **Globuskoordinate** zu erhalten, wird die `GetPointOnTerrainFromScreenCoords` (`double screen_x`, `double screen_y`)-Methode der Google Earth Com API genutzt, die nur normalisierte Bildschirmkoordinaten akzeptiert.

2. Problem

Der Globus muss in ein **Raster** aufgeteilt werden, so dass während einer Aufnahme jede **Globuskoordinate** einer Kachel des Rasters zugeordnet wird – ist die jeweilige Kachel noch nicht instanziiert, wird ein neues Kachel-Objekt erzeugt. Ist die Kachel hingegen schon vorhanden, wird nur eine Zählervariable dieses Kachel-Objekts inkrementiert.

Lösung

Um dem Raster eine dynamische Auflösung zu gewähren, d.h. der Anwender bestimmt, ob eine Heatmap mit geringer, normaler oder hoher Auflösung erzeugt wird, müssen die übergebenen Globuskoordinaten je nach Auflösung abgerundet werden. Die Klasse `Heatmap_Tile` der **Abbildung XVIII** stellt alle nötige Funktionalität zur Verfügung und verwaltet `Heatmap_Tile` Objekte in einer `Hashtable`-Datenstruktur⁽⁶⁷⁾. Ist die Aufnahme beendet, werden alle `Tile`-Objekte en bloc zum SQL Server geschickt.

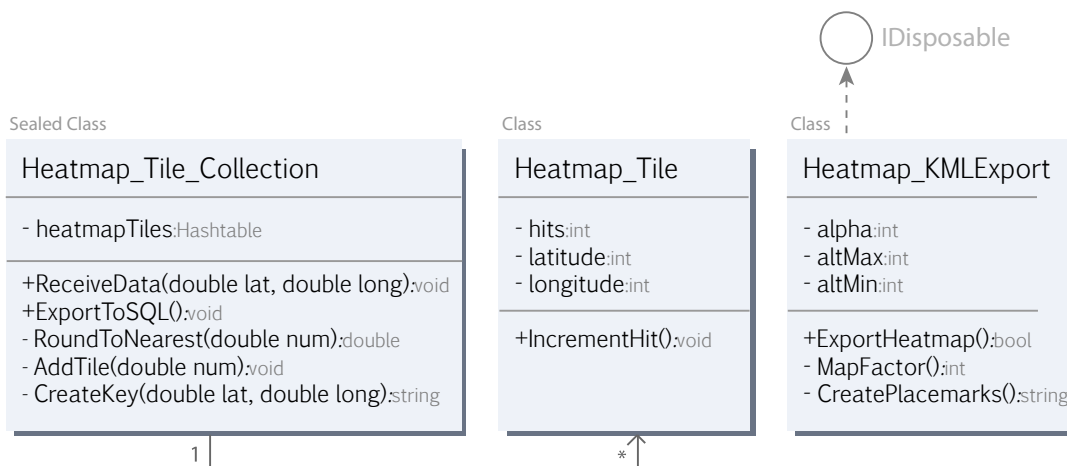


ABBILDUNG XVIII:
Klasse `Heatmap_Tile_Collection` und `Heatmap_Tile`

(67) Eine `Hashtable` wird verwendet, da sich über den erzeugten Schlüssel einer Kachel, der aus `latitude` und `longitude` gebildet wird, gut ermitteln lässt, ob bereits ein Kachel-Objekt diesen Typs in der `Hashtable`-Kollektion vorhanden ist.

3. Problem

Die Verweildauer `hittime` soll über die Farbe und die registrierte Häufigkeit `hits` über die Höhe der entsprechenden Säule visualisiert werden. Der numerische Wert `hittime` als **längste zusammenhängende Verweildauer** soll auf einen Farbwert abgebildet werden, der bei `hittime = 0` (keine Verweilen) mit Blau beginnt und über Gelb und Orange bis hin zu Rot die höchste Verweildauer einer Aufnahme abbildet.

Lösung

Die Klasse `Heatmap_Tile_Export` übernimmt diese Funktionalität. Dazu wird das Intervall an vorhandener `hittime [n,k]` einer Kachel auf das Intervall `[0,255]` projiziert, d.h. der höchste vorkommende Wert wird auf 255 abgebildet. Dieser Wert zwischen 0 und 255 wird nun auf die einzelnen BGR-Komponenten⁽⁶⁸⁾ mithilfe einer der folgenden Funktionen angepasst.

$$f_{(\text{hittime_red})} = \text{hittime} \quad f_{(\text{hittime_green})} = \sin\left(\frac{\pi \cdot \text{hittime}}{255}\right) \cdot 255 \quad f_{(\text{hittime_blue})} = 255 - \text{hittime}$$

Was bei der Rot- und Blaukomponente linear zu vollziehen ist, muss bei der Grünkomponente z.B. durch eine trigonometrische Funktion gelöst werden, da ansonsten nicht für alle Werte die gewünschte Farbe als Ergebnis berechnet würde. Die Abbildung des Wertes `hits` auf die Höhe geschieht ähnlich. **Abbildung XIX** zeigt die Resultate.

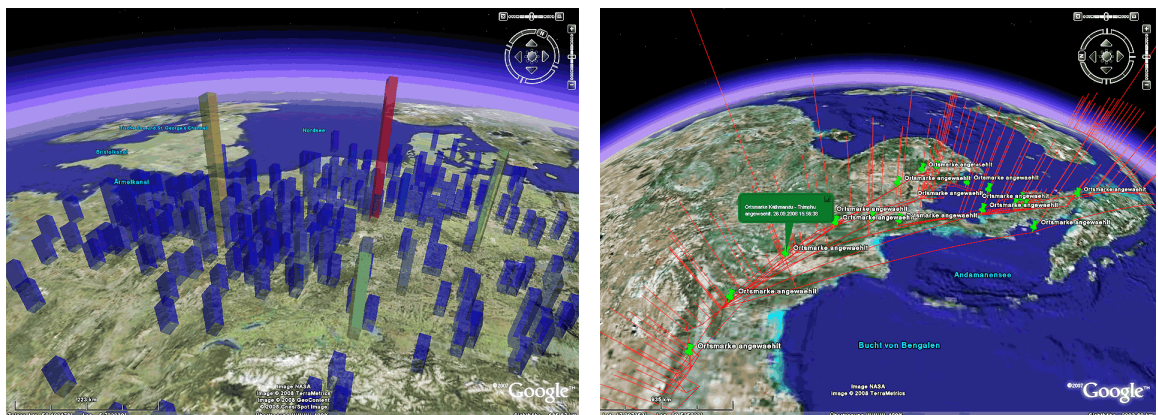


ABBILDUNG XIX:

Exportierte 3D-Heatmap, hohe Auflösung, transparent (links),
Nutzeraktivität, mit Vektoren zur Visualisierung der Kameraentfernung (rechts)
Satellitenbilder © NASA, TerraMetrics, GeoContent, Cnes/Spot Image

(68) Zur Erinnerung: KML verwendet nicht ARGB, sondern das ABGR-Format.

Quantitativ – Feedback

Da das .NET Framework keine Steuerelemente für Diagramme zur Verfügung stellt, ist das Tortendiagramm zur Visualisierung der **durchschnittlichen Nutzerwertung** als Benutzersteuerelement mithilfe von **GDI+** entworfen worden. **Abbildung XX** zeigt diese Komponente in Aktion. Zum Export von Reports ins PDF-Format wurde die freie Klassenbibliothek **iTextSharp**⁽⁶⁹⁾ benutzt, mit der sich nach einem gewissen Einarbeitungsaufwand PDF-Dokumente erstellen lassen. Ein erstellter Report zur Veranschaulichung befindet sich im Anhang.



ABBILDUNG XX:

Durchschnittliche Nutzerwertung mithilfe von GDI+ als Tortendiagramm dargestellt

5.2.2 Realisierung geolyzer – Heuristisch

Um die Modularität an Heuristiken umzusetzen, wird für Nutzer ein Dialogfeld zum Export erschaffen, welches zwei `ListView`-Steuerelemente mit **verfügbaren** und dem Report **zugewiesenen Heuristiken** enthält. Die Klasse `Usability_Heuristic_Export` verarbeitet diese Eingaben, wendet daraufhin die entsprechenden Algorithmen an und exportiert die Ergebnisse als PDF-Datei. Im Paket `Usability` werden die Klassen gemäß **Abbildung XXI** erstellt. Im **Anhang** befindet sich ein Musterreport zur Veranschaulichung.

(69) siehe <http://itextsharp.sourceforge.net/>

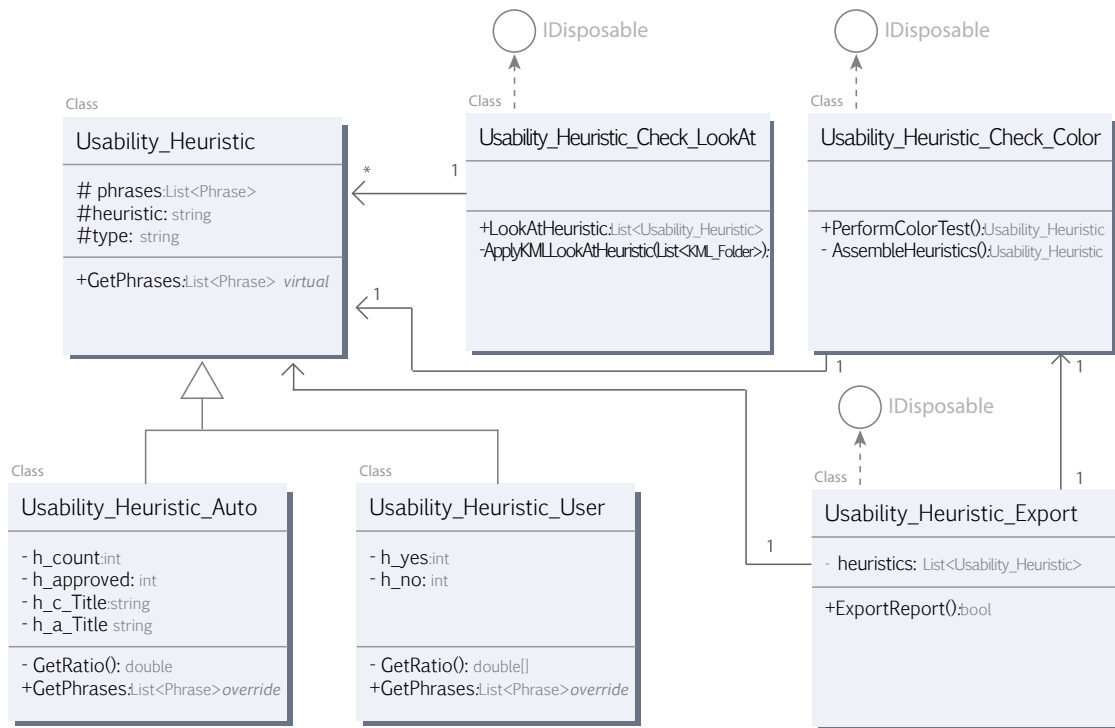


ABBILDUNG XXI:
Klassen zu Usability Heuristiken

- » Usability_Heuristik – Allgemeine Darstellung einer Heuristik mit Attributen zur Beschreibung sowie der virtuellen GetPhrases () -Methode, die die Heuristikergebnisse als passende Absätze für den PDF-Report zur Verfügung stellt.
- » Usability_Heuristik_Auto und Usability_Heuristik_User – Spezialisierung von Usability_Heuristik für automatische und Nutzer-Heuristiken.
- » Usability_Heuristic_Export – Verarbeitet die vom Nutzer hinzugefügten Heuristiken, indem entweder aus dieser Klasse heraus Objekte vom Typ Usability_Heuristic erstellt werden oder dieses durch spezielle Klassen, z.B. Usability_Heuristic_Color, geschieht.
- » Usability_Heuristic_Check_Color – Beinhaltet den Algorithmus zur Überprüfung der Farbkontraste. Erzeugt ein Usability_Heuristic_Auto-Objekt und stellt dieses der Usability_Heuristic_Export-Klasse zur Verfügung.
- » Usability_Heuristic_Check_LookAt – Beinhaltet den Algorithmus zur Überprüfung des Vorhandenseins von LookAt-Tags.

Bei den **Nutzer-Heuristiken** geschieht die Bewertung anhand des Medians zur Zustimmung. Eine Frage wie »*Waren Ortsmarken innerhalb der Präsenz konsistent und wurden wiedererkannt?*« hätte bei einer durchschnittlichen Zustimmung von insgesamt 65% ein mittelmäßiges Ergebnis, d.h im Report würde dieses Ergebnis durch eine orange Färbung dargestellt.

- » **0% - 49%:** rote Darstellung
- » **50% - 79%:** orange Darstellung
- » **80% - 100%:** grüne Darstellung

Die Darstellung durch Farben ist noch keine Usability-Bewertung im engeren Sinn, wie es die **auto-evaluierende Komponente** leisten soll – es dient eher der schnellen Übersicht und zeigt, bei welchen inhaltlichen Aspekten Nutzer am meisten Schwierigkeiten hatten.

Es wurden zwei **automatisch** zu überprüfende **Heuristiken** implementiert⁽⁷⁰⁾. Die Überprüfung des **Farbkontrastes** zwischen **Hintergrund**- und **Textfarbe** ist durch die gesonderte Klasse `Usability_Heuristic_Check_Color` realisiert. Die vom W3C vorgeschlagene Formel zur Berechnung der Farbhelligkeit lautet:

```
double brightness = ((R * 299) + (G * 587) + (B * 114)) / 1000;
```

Sie ist allerdings umstritten, da sie bei einigen Farben, besonders bei Gelbtönen, zu falschen Ergebnissen führt. Im Web hat sich inzwischen eine andere Formel gemäß Pythagoras etabliert, die hier zur Berechnung der Helligkeit hinzugezogen wurde [NBD-TECH].

```
double brightness = Math.Sqrt(R * R * .241 + G * G * .691 + B * B * .068);
```

Die Differenz zweier Farben sollte mindestens 125 betragen. Zum Vergleich: zwei identische Farben ergeben eine Differenz von 0 und Schwarz und Weiß eine Differenz von 255. Der Report listet jegliches Vorhandensein von `Balloon`-Tags mit `bgcolor`- und zugehörigen `textcolor`-Werten sowie der Helligkeitsdifferenz auf.

(70) Die zweite Heuristik beinhaltet das Überprüfen des Vorhandenseins eines LookAt-Tags. Dieses technisch umzusetzen ist trivial und wird nicht gesondert beschrieben.

5.2.3 Realisierung geolyzer – Auto-Evaluierend

Die letzte und gleichzeitig umfangreichste Komponente **geoReport** soll einen PDF-Report nach Art des **Common Industry Formats** generieren, in dem neben den **allgemeinen Daten** zur entsprechenden Sitzung insbesondere **Metriken** zur **Evaluation** der Usability vorhanden sind, die als Einzelwertung in ein Gesamtergebnis einfließen und den sog. **geo-Index** als numerischen Wert von 0 bis 100 bilden.

Da die **CPU-Last** zur Erzeugung des Reports besonders bei umfangreichen Sitzungen hoch ist, wird zuerst ein Dialogfeld zum Export erschaffen, welches als zusätzliches windows-Forms-Element eine **ProgressBar** enthält, die Anwendern die verbleibende Zeit in Balkenform aufzeigt. Hier ist zu beachten, dass der Aufruf **asynchron** ausgeführt wird, um die Anwendung während des Exports nicht zu blockieren. Anstelle eines eigenen Threads eignet sich ein leichtgewichtiger **asynchroner Methodenaufruf**.

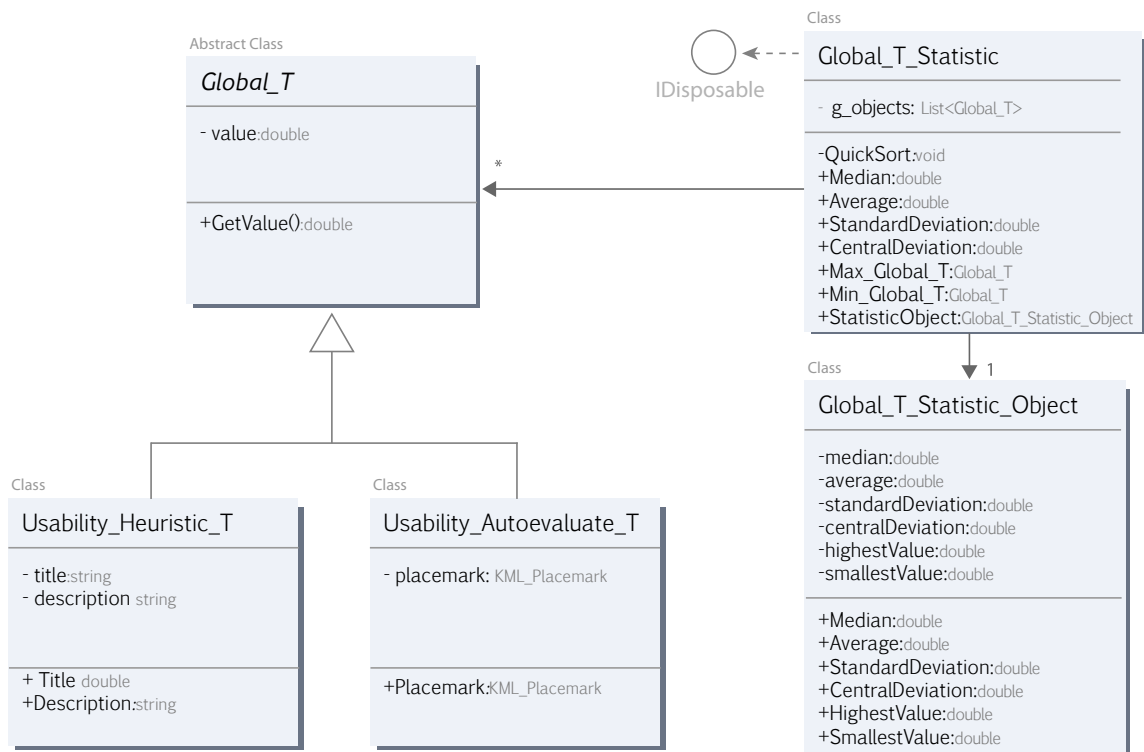


ABBILDUNG XXII:
Klassen zur Erzeugung von Statistiken

Da Statistiken in dieser Komponente eine Rolle spielen, wird das Objekt `Global_T` als allgemeiner Statistiktyp eingeführt, von dem sich spezielle Statistik-Objekte ableiten lassen. Für Objekte diesen Typs wird die Klasse `Global_T_Statistic` erschaffen, in der u.a. die Kenngrößen **arithmetisches Mittel**, **Median** und **Standardabweichung** berechnet werden. Die Ergebnisse lassen sich hier ebenso als leichtes `Global_T_Statistic_Object` erzeugen, falls sie in anderen Objekten, außerhalb des Gültigkeitsbereichs von `Global_T_Statistic`, weiterverwendet werden sollen. **Abbildung XXII** zeigt das Klassendiagramm.

Die Klasse `Usability_Autoevaluate_Report` dient der zentralen Erzeugung der einzelnen **Metadaten** und **Metriken**. Diese Daten werden hier generiert und als Objekte vom Typ `Phrase` der Klasse `PDF_Document` übergeben. **Abbildung XXIII** zeigt die Klassen.

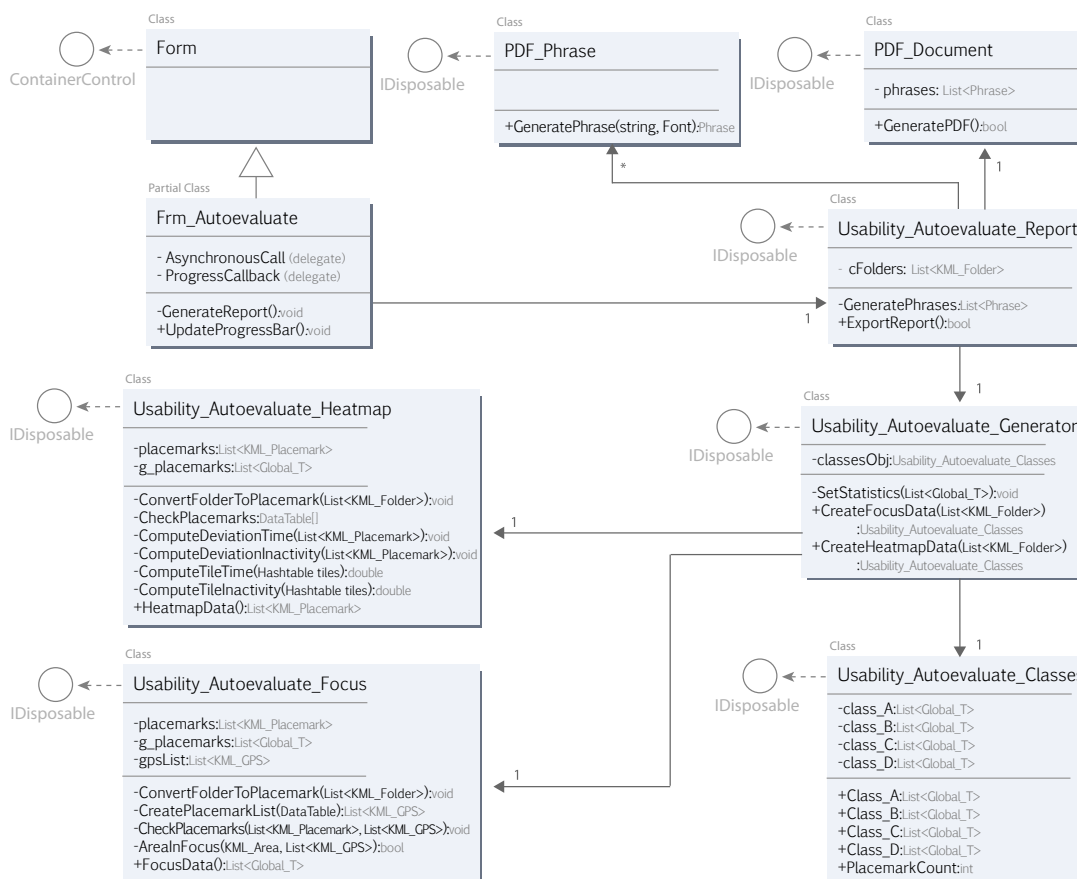


ABBILDUNG XXIII:
Relevante Klassen zur Erzeugung des **geoReports**

Über die Klasse `Usability_Autoevaluate_Generator` werden die Algorithmen in `Usability_Autoevaluate_Heatmap` und `Usability_Autoevaluate_Focus` benutzt, um die erzeugten Metriken gemäß dem Konzept zu klassifizieren. Die Klassifikation geschieht in `Usability_Autoevaluate_Classes`. Im Folgenden werden die beiden relevanten Algorithmen zur Erzeugung der Metriken dargestellt – einmal als Pseudocode und einmal in natürlicher Sprache.

Interpretation Kamerafokus

Algorithmen zur Evaluation, ob Regionen um Ortsmarken sich mit dem aufgezeichneten Verlauf des Kamerafokus schneiden – und die Ortsmarke folglich betrachtet wurde.

```
Algorithm FocusData() returns List<Global_T>
    query: Ortsmarken as List<KML_Placemark>;
    query: Aufnahmen as DataTable;
    foreach Aufnahme in Aufnahmen do
        query: GPS-Koordinaten as List<KML_GPS>;
        call: CheckPlacemarks (GPS-Koordinaten, Ortsmarken);
    return Ortsmarken as List<Global_T>;
```

```
Algorithm CheckPlacemarks (GPS-Koordinaten as List<KML_GPS>, Ortsmarken as
List<KML_Placemark>) returns nothing
    foreach Ortsmarke in Ortsmarken do
        new: Region as KML_Area from Ortsmarke;
        if call: AreaInFocus (Region, GPS-Koordinaten) then
            Ortsmarke.ErhöheZähler();
```

```
Algorithm AreaInFocus (Region as KML_Area, GPS-Koordinaten as List<KML_GPS>)
returns boolean
    foreach GPS-Koordinate in GPS-Koordinaten do
        if Region.EnthältKoordinate (GPS-Koordinate) then
            return true;
    return false;
```

Interpretation Heatmap

Algorithmen zum Betrachten von Heatmap-Kacheln innerhalb Regionen um Ortsmarken. Berechnung der Gesamtzeit sowie der Verweildauer innerhalb dieser Region. Die relevanten Methoden werden kurz erläutert.

```
public List<KML_Placemark> HeatmapData() { };
    » Erzeugt DataTable[] mit Heatmap-Tupeln zu allen Aufnahmen
    » Ruft CheckPlacemarks() auf und gibt analysierte Ortsmarken als Liste zurück

private void CheckPlacemarks(DataTable[] table) { };
    » Durchläuft alle Ortsmarken der Sitzung (äußere Schleife)
    » Erzeugt für jede Ortsmarke eine KML_Area (aktive Region)
    » Durchläuft alle Aufnahmen des DataTable[] (1. innere Schleife)
    » Durchläuft alle Tupel der Aufnahme (2. innere Schleife)
    » Erzeugt aus Geodaten des Tupels ein KML_GPS-Objekt
    » Vergleicht, ob KML_GPS-Objekt innerhalb KML_Area liegt
    » Wenn ja, dann aus dem KML_GPS-Objekt einen Hash-Key erzeugen
    » Existiert Hash-Key in Hashtable, dann gefundenes Objekt nehmen und
      Heatmap-Daten aufsummieren. Ansonsten aus dem KML_GPS-Objekt mit den
      weiteren Heatmap-Daten des Tupels ein neues Heatmap-Tile Objekt er-
      zeugen und in der Hashtable ablegen
    » Außerhalb der Schleifen für alle Heatmap-Tile Objekte der Hashtab-
      le die Gesamtzeit sowie die Inaktivität für die jeweilige Ortsmarke
      berechnen und diese Daten dem Ortsmarken-Objekt hinzufügen
```

Die Methoden `ComputeDeviationTotalTime` und `ComputeDeviationInactivity` berechnen für die einzelnen Ortsmarken nun mithilfe der bereits genannten Statistik-Klassen die Werte zur **Abweichung** des **Erwartungswerts** von der **Standardabweichung**. Zur Wiederholung: Die Evaluation der Mausinaktivität basiert darauf, für jede Aufnahme das **typische Nutzerverhalten** als **Median** der Inaktivität zu berechnen. So kann für jede Ortsmarke innerhalb der Aufnahme überprüft werden, ob einzelne Inaktivitäten vom typischen Nutzerverhalten abweichen. Als **Toleranzgrenze** wird die **Standardabweichung** σ benutzt, die bestimmt, ob die Abweichung **signifikant** ist oder nicht.

5.3 Testdurchläufe unter Realbedingungen

Um die Praxistauglichkeit des **geolyzer** zu evaluieren wurden zwei Testreihen durchgeführt. Die erste Testreihe mit der **geolyzer** Version 1.15 wurde in Kalenderwoche 39 beim Bremer Start-Up *nuterra* durchgeführt. Die letzte Testreihe wurde am sechsten Oktober 2008 bei der visuamundo GbR in Oldenburg vollzogen. Hier kam die finale **geolyzer** Version 1.2 zum Einsatz, mit der sich evaluierende Reports erstellen lassen.

5.3.1 Nutzertest und Evaluation nuterra



Das Bremer Start-Up *nuterra*⁽⁷¹⁾ existiert seit dem März 2008 und hat sich auf Inszenierungen in Geobrowsern in Zusammenarbeit mit dem Gestaltungspartner visuamundo GbR spezialisiert. Gefördert wird *nuterra* durch ein EXIST-Gründerstipendium durch das Bundesministerium für Wirtschaft und Technologie (BMWi).

Es hat sich angeboten, gleichzeitig die Software **geolyzer** seitens von *nuterra* evaluieren zu lassen und einen ersten Usability-Test mit Probanden zu in der Praxis verwendeten KML-Inhalten durchzuführen. Als vorbereitender Aufwand wurden insgesamt drei Tage benötigt, um einmal die Software vor Ort zu installieren und zu konfigurieren, das Layout der KML-Inhalte anzupassen⁽⁷²⁾, geeignete Inhaltsfragen zu definieren und einen zusätzlichen Fragenkatalog zu erschaffen, mit dem Probanden u.a. das Werkzeug **geolyzer** an sich bewerten sollen. Der Test wurde mit fünf Probanden durchgeführt und die jeweiligen Aufnahmen hatten eine Dauer von 15-20 Minuten.

Der Test wurde mit **drei inhaltlich identischen** KML-Dateien durchgeführt, die sich nur in der Navigation unterscheiden. Der Titel der KML lautet »Weltreise« und beschreibt inhaltlich die wichtigsten Stationen einer Reise mit einem VW Käfer von Hamburg nach Australien.

- » **KML 1:** Test komplett **ohne Navigation**
- » **KML 2:** Test mithilfe der Google Earth **Ordernavigation**
- » **KML 3:** Test mit **geführter Navigation**

(71) *nuterra*, c/o TZI, Am Fallturm 1, 28359 Bremen, <http://www.nuterra-bremen.de/>

(72) *nuterra* entwickelt auf Macintosh-Systemen. Google Earth stellt KML-Inhalte auf Mac-Systemen anders dar als auf Windows-Systemen.

Ergebnisse Nutzertest

Die verwendete **geolyzer** Version 1.15 unterstützt **Tracking** sowie den **Probandenreport** – die heuristischen und auto-evaluierenden Komponenten sind noch nicht enthalten.

Beim Betrachten der Ergebnisse fällt allgemein auf, dass vier der Probanden recht ähnlich navigieren und einen fast identischen Fokus-Verlauf erzeugen, der annähernd synchron zur Strecke der Weltreise verläuft. Der fünfte Proband hingegen zeigt deutliche Abweichungen und fokussiert oft Regionen auf dem Globus, die mit den Inhalten nicht in Verbindung stehen. Wichtigstes zusammenfassendes **Ergebnis** der Tracking-Funktion **Nutzeraktivität** ist, dass bei Aufnahmen zur **KML 1** (*ohne Navigation*) deutlich mehr Ortsmarken überhaupt nicht beachtet werden und die Abweichungen zum Reiseverlauf höher sind als bei Aufnahmen mit geführter Navigation.

Ebenso interessant ist das **Ergebnis** der Tracking-Funktion **Heatmap**. Generell gilt, dass in Regionen um Ortsmarken die höchste Dichte an Heatmap-Kacheln besteht. Einige Ortsmarken haben durchschnittlich sehr hohe Zugriffs- und Zeitwerte, bei einigen sind die Werte eher unter dem Durchschnitt. Mit der finalen Version 1.2 können diese Beobachtungen mit greifbaren numerischen Werten verglichen werden.

Der **Probandenreport** gibt Auskunft über Personendaten sowie allen abgegebenen Meinungen zu vordefinierten Fragen. Zusammenfassendes **Ergebnis**: Probanden bewerten nicht einheitlich und die gesamte Bandbreite von sehr guten Bewertungen der KML-Inhalte über eher neutralen Wertungen bis hin zu schlechten Bewertungen ist vertreten. Eine ausführliche Auflistung aller Ergebnisse zum Nutzertest befindet sich im **Anhang**.

Evaluation geolyzer

Die Resonanz zum **geolyzer** ist durchweg positiv. Das Programm wird von *nuterra* als sehr umfassend bezeichnet, mit dem sich ferner gut arbeiten lässt. Dennoch gibt es auch eine Reihe an Mängeln, Verbesserungsvorschlägen sowie Wünschen für künftige Versionen. Eine ausführliche Darstellung dieser Punkte befindet sich im **Anhang**. Von den Probanden wird die Handhabung des **geolyzer** vier Mal als *sehr gut* oder *eher gut* bezeichnet, einmal wird sie als *eher schlecht* bewertet.

5.3.2 Nutzertest visuamundo



Die visuamundo GbR⁽⁷³⁾ existiert seit 2007 und hat sich auf Inszenierungen im Geotainment-Bereich spezialisiert. Für den letzten Test kam die finale **geolyzer** Version 1.2 zum Einsatz, um durch zwei exemplarische KML-Dateien die Auto-Evaluation in der Praxis zu testen. Es wurden zwei Sitzungen erstellt, die mit jeweils drei Probanden getestet wurden (es gibt somit insgesamt sechs Aufnahmen). Bei den KML-Inhalten handelt es sich um ein imitiertes Reiseportal der TUI mit aktuellen Reiseangeboten.

KML 1: *Reiseportal der TUI (nicht optimal aufbereitet)*

- » nicht einheitliche Icons sowie *keine* Menünavigation
- » Texte und Farben recht unfreundlich formatiert und überlappende Inhalte
- » keine Zurückfunktion
- » dem Nutzer wird nicht verdeutlicht, wo er sich gerade befindet
- » Sinn und Zweck der Präsenz wird in der Start-Ortsmarke nicht vermittelt

Die **KML 2** besteht aus denselben Inhalten und ist besser aufbereitet, d.h. sie leidet nicht an den genannten Mängeln der **KML 1**. Sie ist allerdings nicht mit professionellen Inszenierungen zu vergleichen – sie dient lediglich als Kontrast zu den absichtlich hinzugefügten Mängeln der **KML 1**. Zur Verdeutlichung: Mit diesem zweiten Usability-Test sollen nicht primär die Inhalte der exemplarischen KMLs bewertet werden – es geht darum, die Funktionalität der Auto-Evaluation auf Korrektheit zu überprüfen und den erstellten Report mit den Beobachtungen während des Tests zu vergleichen.

Testergebnisse

Für beide Sitzungen wurde ein **geoReport** nach Art des CIF exportiert. Auszüge des Reports sind im Anhang zu finden. Im Folgenden werden nur die Metriken des Reports zusammengefasst. Um das Ergebnis vorwegzunehmen: die **KML 1** hat mit einem **geoIndex** von 41 Punkten schlechter abgeschnitten als **KML 2** mit 68 Punkten.

Die gestellten **Nutzer-Aufgaben** zu den Inhalten waren leicht zu erledigen, so dass in beiden Sitzungen eine *Completion Rate* von 100% erreicht wurde. Bei den Nutzer-Heuristiken gab es bei **KML 2** eine durchschnittliche Zustimmung von 60.71% und bei der mangelhaften **KML 1** immerhin noch 49.25% durchschnittliche Zustimmung. Das Ergeb-

(73) visuamundo GbR, Philosophenweg 16, 26121 Oldenburg, <http://www.visuamundo.com>

nis der **Farbkontrast-Überprüfung** zeigte bei der **KML 1** korrekt an, dass nur bei einer von insgesamt 16 Ortsmarken die gebrauchstaugliche Farbgebung gegeben ist. Bei **KML 2** wurden alle Farbkontraste der Ortsmarken als geeignet deklariert. Bei beiden KML-Inhalten existieren insgesamt 13 von 16 LookAt-Tags – auch dies wurde im Report richtig vermerkt.

Die Analyse des **Kamerafokus** zeigte, dass bei der **KML 2** mit durchschnittlich 6.34 von 16 Ortsmarken insgesamt mehr Ortsmarken als bei der **KML 1** mit durchschnittlich 5.11 von 16 Ortsmarken fokussiert wurden – im Gegensatz zur **KML 1** war es hier durch ein übersichtliches Startmenü möglich, alle Ortsmarken direkt anzufliegen. Das Ergebnis deckt sich mit der subjektiven Beobachtung der Probanden.

Die Ergebnisse zur Auswertung der **Gesamtzeit** und der **Inaktivität** sind für diesen Test von großer Relevanz. Auch wenn bei fast allen Ortsmarken aufgrund fast identischer Inhalte erwartungsgemäß dieselben Zeitwerte vorhanden sind, so zeigt der Report bei der Ortsmarke »Atlantico Golfe Hotel« deutliche erhöhte Abweichungen. Die betreffende Ortsmarke enthält absichtlich irreführende Inhalte und befindet sich mitten im Atlantik. Zwei von drei Probanden zeigten eine Verunsicherung und ein Ruhen des Mauszeigers von bis zu 20 Sekunden. Im Report wird dies korrekt wiedergegeben. Anhand diesen Beispiels wurde gezeigt, dass es mit **geolyzer** prinzipiell möglich ist, zeitliche Anomalien im Nutzerverhalten festzuhalten.

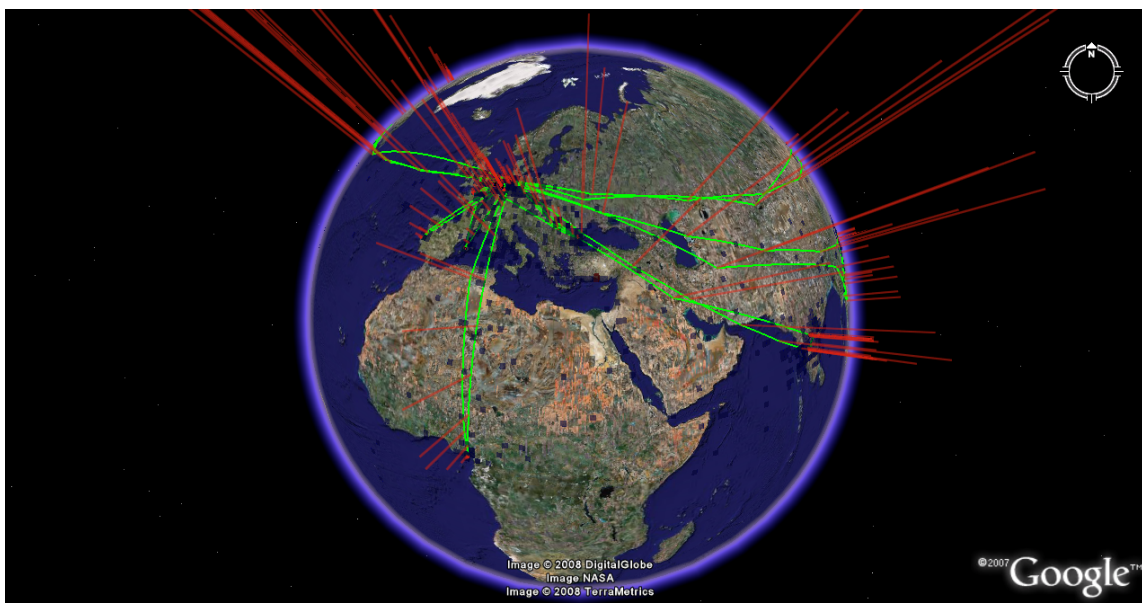


ABBILDUNG XXIV:
Exportierte 3D-Heatmap und Nutzeraktivität eines Probanden
Satellitenbilder © NASA, DigitalGlobe, TerraMetrics

“Turn on, tune in, drop out.”

Dr. Timothy Leary
American Psychologist and Icon of 1960s Counterculture

6. Diskussion

Mit dem Ende der Diplomarbeit ist das erreicht, was anfangs nur eine vage Idee war. Ein softwarebasiertes Usability-Tool für Geobrowser-Inhalte auf dem Präsentierteller. Ressourcenschonend, einfach in der Handhabung und innovativ. Natürlich ist nicht alles Gold, was glänzt, und darum gilt es, sich von der tendenziös-euphorischen Sichtweise während des Entwurfs zu befreien, zurückzulehnen und einen ernsten objektiven Blick zu werfen.

Mit diesem Kapitel soll der Fokus auf das System geolyzer erstmalig von innen nach außen verlagert werden. Was wurde an vorgenommenen Zielen erreicht – und wie lässt sich das Erreichte beurteilen? An welchen Schwächen leidet das System und wie müssten diese in künftigen Versionen kuriert werden?

6.1 Evaluation geolyzer – Quantitativ

Es wurden alle Komponenten, die im Konzept genannt wurden, auch annähernd vollständig umgesetzt. Einige Abstriche waren bei den Heuristiken aus zeitlichen Gründen nicht zu vermeiden. Und das auch nur dort, wo ein relativ geringer Nutzwert einem unverhältnismäßig hohen Aufwand gegenüberstand. Quantitative und auto-evaluierende Funktionen wurden vollständig realisiert. Der Soll-Zustand ist faktisch erreicht.

Mit dem **geolyzer**-Tracking ist eine Alternative zu klassischen Verfahren wie z.B. Screen-capturing erschaffen worden. Es ist möglich, alle relevanten Aktionen des Nutzers **orts-** und **zeitgenau** wiederzugeben. Der Verlauf des **Kamerafokus** zeigt, wo und wann Probanden den Test begonnen haben und welche Regionen sie während des Tests fokussierten. Ortsmarken mit Nutzerereignissen, wie. z.B. das Ausführen der Suchfunktion, sind genau an den Punkten des Verlaufs mit einem Zeitstempel platziert, wo sie auch tatsächlich stattgefunden haben. Mit der Möglichkeit, alle Aufnahmen einer Sitzung in *eine* KML-Datei zu exportieren, können Analysten sofort erkennen, welche Ortsmarken von den Probanden überwiegend wahrgenommen wurden und welche nicht. Auch Abweichungen in der Navigation einzelner Probanden sind auf Anhieb zu erkennen. Die Funktionalität, eine beliebige Menge an Aufnahmen gleichzeitig in *einer* Datei darzustellen, ist beim Screencapturing prinzipiell nicht vorhanden.

Die **3D-Heatmaps** zeigen Analysten durch die Höhe der einzelnen Säulen bzw. Kacheln, wo der Mauszeiger **wie oft** insgesamt und darüberhinaus durch die Farbe der Säule, **wie lange** der Mauszeiger dort hintereinander registriert wurde. Durch Letztgenanntes ist es prinzipiell möglich, ab einer ausreichend geringen Kachelgröße in Abhängigkeit zur Entfernung der Kamera, die Behauptung aufzustellen, der Mauszeiger ruhte hier für n Sekunden. Insgesamt bietet die 3D-Heatmap eine Funktionalität, die beim 2D-Korrelat für Web-Analysen nicht existiert. Dort kann durch Farbdarstellung *entweder* die absolute Häufigkeit *oder* die Verweildauer des Mauszeigers visualisiert werden – aber nicht beides zugleich in einer Heatmap.

Der Probandenreport als letzte quantitative Funktion ist ein nützliches Werkzeug, um schnellen Überblick über abgegebene Meinungen aller Probanden zu erhalten. Trotz des Erfolgs der implementierten Funktionalität existieren auch **Schwächen**, die im Folgenden genannt werden und Ergebnisse unter bestimmten Umständen verfälschen.

Ein Problem stellen sog. **Phantomverläufe**. Diese treten dann auf, wenn Probanden in der linken Ordnerstruktur eine weit genug entfernte Ortsmarke anwählen. Google Earth wechselt (*bzw. »fliegt«*) rasch zu dieser Ansicht und **geolyzer** speichert die nächste Fokuskordinate. Beim Export werden nun die beiden jeweiligen Koordinaten durch das Vektorobjekt `LineString` verbunden. Liegt dieser Phantomverlauf in der Nähe oder auf einer anderen Ortsmarke, so entsteht der Eindruck, der Proband hätte diese Ortsmarke fokussiert, was nicht der Fall ist. Immerhin geben die beim Export generierten Ortsmarken zu den Nutzerereignissen Auskunft, ob Ortsmarken in der Ordnerstruktur angewählt wurden. So lässt sich erkennen, ob die einzelnen Strecken des `LineString` anhand manueller Navigation oder aufgrund des Anklickens einer Ordnerstruktur-Ortsmarke entstanden sind.

Eine weitere Schwäche stellt die derzeit maximale **Auflösung** der **3D-Heatmaps** dar. Die Auflösung lässt sich zwar im Programm vom standardmäßigen Wert `1.0` (*normal*) bis maximal `0.25` (*hoch*) anpassen⁽⁷⁴⁾, was einer Kachel von einem Viertel Längen- und Breitengrad entspricht⁽⁷⁵⁾ – dennoch gibt es Fälle, wo dies eben nicht ausreicht. Das wäre der Fall bei Inhalten, die sehr nah beieinander liegen. Da es im zeitlichen Rahmen der Arbeit nicht mehr möglich war, muss für künftige Versionen analysiert werden, ob eine Auflösung von `0.1` bis hin zu `0.01` nicht zu viele Vektorobjekte und Einträge in der Datenbank erzeugt. Theoretisch würden bei einer Auflösung von `1.0` – gesetzt den unwahrscheinlichen Fall, dass der Mauszeiger tatsächlich auf allen Kacheln des gesamten Globus registriert wird – maximal `64.800` Tile-Objekte in Programm und ebenso viele Tupel in der Datenbank je Aufnahme erzeugt. Bei `0.25` wären es maximal `460.800` und bei `0.01` schon problematische `648.000.000` Objekte.

(74) Eine höhere Auflösung wäre nur durch Änderungen im Source Code machbar.

(75) Zum Vergleich: Mit einem Wert von `0.25` Lat. und Long. wird eine Fläche fast so groß wie Hamburg abgedeckt.

6.2 Evaluation geolyzer – Heuristisch

Es wurden **Nutzer-Heuristiken** sowie zwei **automatisch** überprüfbare **Heuristiken** umgesetzt. Bei den automatisch zu validierenden Heuristiken handelt es sich durchweg um Funktionen zur Validierung des Markups. Hier wurden nur die zwei umgesetzt, die zum aktuellen Zeitpunkt auch einen reellen Nutzen bieten. Dabei handelt es sich einmal um die Überprüfung des Farbkontrastes zwischen Hintergrund- und Textfarbe des Ortsmarken-Balloons – um Probleme für visuell eingeschränkte Nutzer zu minimieren – und weiterhin um die Überprüfung des Vorhandenseins eines `LOOKAT`-Tags. Dieses Tag sollte für alle Ortsmarken vorhanden sein, um die aus dem Designprozess als geeignet definierte Kameraorientierung dem Inhalt zuzuweisen. Die beiden anderen im Konzept speziell für Geobrowser-Inhalte definierten Heuristiken waren entweder indirekt umgesetzt (*wohlgeformtes XML*) oder wurden aufgrund diffusem Nutzwert⁽⁷⁶⁾ nicht realisiert. Generell leidet das Konzept an zu wenig automatisch überprüfbaren Heuristiken speziell für Geobrowser-Inhalte. Hier müsste weiter geforscht werden, welche Heuristiken sich ferner zur Überprüfung eignen.

Die anderen bislang nicht umgesetzten Heuristiken beziehen sich auf die Barrierefreiheit, was allerdings nicht bedeutet, dass dieser Aspekt für Geobrowser weniger wichtig ist. Im Gegenteil, gerade hier dürfen wie beim normalen Web die Bedürfnisse der Menschen mit Einschränkungen nicht vernachlässigt werden. Bedauerlich ist, dass Google erst jetzt damit anfängt, dem Thema Barrierefreiheit im Geobrowser mehr Beachtung zu schenken. Zum aktuellen Zeitpunkt hätte die Realisierung von Algorithmen, die das KML-Markup auf Einhaltung von **Accessibility-Richtlinien** hin überprüfen – z.B. dem Vorhandensein von alternativen Text – wenig praktischen Nutzen. HTML-Entitäten werden noch zu wenig unterstützt und Bildschirmlesegeräte wären beispielsweise nicht in der Lage, alternativen Text innerhalb von Ortsmarken-Inhalten korrekt wiederzugeben.

Durch die **Nutzer-Heuristiken** wird der Prozess der Nutzerbefragung beschleunigt. Diese Heuristiken werden mit dem Programm installiert und können insgesamt oder als Selektion jeder Sitzung zugewiesen werden. Ab einer ausreichenden Menge an Aufnahmen hat der Durchschnittswert der Zustimmung einzelner Fragen eine Aussagekraft. So schafft die Auswertung via übersichtlichem Heuristik-Report schnell Auskunft, ob Standards eingehalten wurden und inwieweit die Nutzer Fragen mit heuristischen Inhalten zustimmen.

(76) Damit ist die Validierung der Anzahl an Folder-Tags gemeint. Die Begründung, warum dies nicht implementiert wurde, befindet sich im Konzept.

6.3 Evaluation geolyzer – Auto-Evaluierend

Gemäß dem Konzept sollte mit dieser Komponente das realisiert werden, was allgemein noch kaum erprobt ist: die automatische Bewertung von Inhalten hinsichtlich der Gebrauchstauglichkeit auf Basis rechnergestützter Nutzereingaben. Völlig ohne Probanden ist die Evaluation auch weiterhin nur in Grenzfällen möglich, die sich auf einen Teilbereich der Usability beschränken, z.B. bei der Überprüfung des Markups. Es war auch nicht das Ziel, einen Algorithmus zu entwickeln, der Inhalte autark und ohne Probanden bewertet. Es ging viel mehr darum, die Aktionen der Probanden, die im Geobrowser aufgezeichnet wurden, automatisch aus- und bewerten zu lassen.

Und mit der auto-evaluierenden Komponente ist dies weitestgehend gelungen: automatisch auszuwerten, ob Inhalte vom Probanden fokussiert wurden – das war bislang nur mit einem Eye-Tracking System machbar. Und durch die aufgezeichneten Heatmap-Daten herauszufinden, wie oft und wie lange am Stück ein Inhalt vom Probanden begutachtet wurde, ist im Sinne der Usability ein mächtiges Instrument – besonders wenn die erhaltenen Daten mit vielen anderen Daten einer Sitzung automatisch kombiniert und verglichen werden.

Der so genannte **geolIndex** als abschließendes Reportergebnis gibt einen schnellen Überblick über die Qualität der Inhalte. Auch wenn dieser durch eine längere Praxisphase noch genauer untersucht und angepasst werden muss, so liefert er bereits jetzt einen greifbaren Hinweis, ob Inhalte zum Gebrauch taugen oder nicht.

Verständlicherweise ist einiges aufgrund der kurzen Realisierungszeit noch mit **Schwächen** behaftet. Die genannten **Phantomverläufe** und die Problematik der **Auflösung** sind zwei Beispiele, die auto-evaluierende Ergebnisse verfälschen können. Ein drittes Beispiel ist die noch unvollkommene **Schnittpunktberechnung** zwischen Fokuspunkt und `KML_Area`, die aufgrund knapper Zeit vorerst trivial implementiert wurde. Liegen zwei zusammengehörige Fokuspunkte in einigen Fällen zu weit auseinander und die `KML_Area` genau dazwischen, so gilt die Area fälschlicherweise als nicht fokussiert. Hier sollte in künftigen Versionen berechnet werden, ob der Vektor zweier zusammengehöriger Fokuspunkte irgendwo die `KML_Area` schneidet, was generell zu exakteren Ergebnissen führt.

Fazit

Allgemein betrachtet bleibt das größte Problem beim **Tracking** die **Auflösung**, und zwar dann, wenn Inhalte sehr nah beieinander liegen. Für Inszenierungen, die beispielsweise auf einzelne Stadtteile einer Großstadt beschränkt sind, liefert das Tracking für die **Auto-Evaluation** keine verwertbaren Ergebnisse. Einerseits liegen mehrere Ortsmarken innerhalb der aktiven Region (*Sichtbereich des Probanden bei höchster Zoomstufe*), andererseits beherbergen auch einzelne Kacheln der Heatmap mehrere Ortsmarken, so dass die exakten Zeitwerte für die jeweiligen Ortsmarken nicht ermittelt werden können.

Weiterhin ist eine generelle Schwierigkeit beim Geobrowser der **Abstand der Kamera zur Erdoberfläche**. Um uneingeschränkt exakte Ergebnisse zu erhalten, sollte stets bei *allen* Tracking-Verfahren der Wert des `FocusRange` mit einbezogen werden. So müsste beispielsweise die `KML-Area`⁽⁷⁷⁾, die die aktive Region einer Ortsmarke definiert, zwingend vom Abstand der Kamera zur Globusoberfläche abhängen. Die Praxistests haben ergeben, dass Probanden in der Regel mit einem ähnlichen Abstand zwischen 50 und 200 Kilometern navigieren. Und hier liefern die implementierten Algorithmen brauchbare Ergebnisse. Das ändert sich aber bereits, wenn Nutzer aus *sehr großer* oder *sehr geringer* Entfernung navigieren – hier müsste die `KML-Area` dynamisch angepasst werden, d.h. sie müsste sich bei großen Entfernungen vergrößern und bei einer Navigation sehr nah an der Oberfläche entsprechend stark verkleinern. In Kombination mit der Auflösung wird ein sehr geringer Abstand besonders bei der zusammenhängenden Verweildauer des Mauszeigers zum Problem – in diesem Fall wird das Ergebnis zur Inaktivität deutlich verfälscht.

Zusammenfassend lässt sich sagen, dass Konzept und Realisierung der Usability-Komponenten eine Bereicherung zur Untersuchung und Verbesserung von Geobrowser-Inhalten hinsichtlich der Gebrauchstauglichkeit darstellen. Für einen ersten Modellversuch sind die Ergebnisse durchaus erfreulich, wenngleich an den genannten Schwächen natürlich noch geforscht und verbessert werden muss. Besonders hervorzuheben ist, dass die im letzten Praxistest verwendete schlecht aufbereitete KML auch im Report schlechter abgeschnitten hat als die KML mit denselben Inhalten in optimal aufbereiteter Form.

(77) Zur Wiederholung: Die `KML_Area` dient dazu, eine aktive Region als Sichtbereich des Probanden um Ortsmarken zu definieren. Durch sie kann verglichen werden, ob Fokuspunkte oder Heatmap-Kacheln innerhalb dieser aktiven Region liegen – und somit zur Ortsmarke gezählt werden oder nicht.

“The good old times may have been good in their way, though their goodness is generally exaggerated; but to attempt to revive their ways of thought for the use of later generations is like putting new wine into old bottles.”

Prentice Mulford
19th Century American Journalist and Mystic

7. Zusammenfassung und Ausblick

Nimmt man wahr, wie Pflege und Erweiterung der Google Earth COM API seitens Google vernachlässigt und gleichzeitig die API des Browser Plugins lanciert wird, so wird deutlich, dass die Zukunft nicht der Client, sondern viel eher die Browser-Version ist. Mit der Client-Version aufs falsche Pferd gesetzt?

Nein, denn zum Zeitpunkt des Konzepts und der ersten Realisierung existierte noch kein Browser-Plugin. Des Weiteren lässt sich das Realisierte auch ebenso auf das Plugin abbilden. Sollte dieses einmal erwartungsgemäß genauso mächtig sein wie die Client-Version, eröffnen sich hier sogar neue Perspektiven.

Dieses letzte Kapitel gibt zusammenfassend Überblick über Gewolltes, Erdachtes und Erreichtes und stellt die Arbeit in einen größeren Zusammenhang. So wird ein Blick nach vorn gewagt, um zu sinnieren, wie die Zukunft des geolyzer aussehen könnte bzw. was mit den Ergebnissen auch außerhalb des geolyzer-Mikrokosmos theoretisch machbar wäre.

Das Ziel dieser Arbeit war die Konzeption und Entwicklung eines ersten softwarebasierten Werkzeugs zur Untersuchung und Verbesserung der Gebrauchstauglichkeit von Inhalten im Geobrowser Google Earth. Als Fundament zur Realisierung wurde der existierende Prototyp **geolyzer** benutzt. Es wurde analysiert, welche gängigen Usability-Verfahren sich auf den Bereich der Geobrowser übertragen lassen und welche noch nicht vorhandenen Verfahren speziell für diesen Bereich infrage kommen und zu definieren sind. Als Ergebnis der Arbeit können Usability-Tests für Geobrowser-Inhalte durch **geolyzer** weitestgehend automatisiert werden. Neben Aufzeichnungs- und Visualisierungsfunktionen zum Nutzerverhalten wurde Funktionalität erdacht und realisiert, die aufgezeichnetes Nutzerverhalten automatisch analysiert, evaluiert und die Ergebnisse als übersichtlichen Report zur Verfügung stellt.

Als Einstieg ins Thema wurde dargelegt, dass Geobrowser im Web des 21. Jahrhunderts eine bedeutsame Rolle spielen und aller Wahrscheinlichkeit nach Auswirkungen auf Soziales und Kulturelles ausüben werden. Diese noch recht junge Technologie hat ihr volles Potenzial bis dato noch nicht ausgeschöpft und eine der primären Schwächen ist noch immer die mangelnde Gebrauchstauglichkeit, die einerseits für das Werkzeug an sich und ebenso für veröffentlichte Inhalte gilt. Als eine der Ursachen wurden die Unterschiede hinsichtlich der Navigation zwischen Web- und Geobrowsern genannt.

Darauf aufbauend wurden erprobte Konzepte und Verfahren zur Gebrauchstauglichkeit analysiert, um auf dieser Basis die ersten Schritte in Richtung passendes Usability-Konzept für Geobrowser zu setzen. Auch das Thema Barrierefreiheit wurde, soweit dies für Geobrowser möglich war, mit in die Betrachtung einbezogen. Als Ergebnis wurde festgestellt, dass mit Ausnahme der qualitativen Verfahren sich quantitative und heuristische Usability-Verfahren eignen, um sie in den Prototyp **geolyzer** zu integrieren. Es wird zum ersten Mal die Idee eines neuen, auto-evaluierenden Verfahrens für Geobrowser-Inhalte erwähnt.

Im Kapitel zum existierenden Prototypen **geolyzer** wurde der Ist-Stand sowie die verwendete Technologie erläutert. Des Weiteren wurde der Soll-Zustand definiert, der aus Tracking- und Visualisierungsfunktionen des Nutzerverhaltens, heuristischer Validierung sowie auto-evaluierender Funktionalität zur automatischen Bewertung von Geobrowser-Inhalten besteht.

Im Realisierungsteil wurde ein Konzept zu quantitativen, heuristischen und auto-evaluierenden Verfahren für Geobrowser erdacht, die für **geolyzer** fast vollständig umgesetzt wurden. Zur Evaluation der Praxistauglichkeit wurde **geolyzer** in Zusammenarbeit mit zwei Unternehmen aus dem entsprechenden Bereich getestet. Die durchweg positive Resonanz der Unternehmen zeigte, dass ein tatsächlicher Bedarf an einem Werkzeug dieser Art besteht. Auch die Ergebnisse der Probandentests zeigten, dass mit dem Werkzeug sehr wohl qualitative Aussagen zur Gebrauchstauglichkeit möglich sind. Gleichzeitig haben die Tests auch Grenzen und Schwächen des Systems **geolyzer** offenbart – gewisse Umstände können Ergebnisse verfälschen, so dass bei einigen Features noch Verbesserungsbedarf besteht.

Zusammenfassend ist ein Werkzeug entstanden, welches die gesteckten Ziele, wenngleich nicht zu 100%, so immerhin weitestgehend erreicht hat. Neue Ansätze und Richtlinien zum gebrauchstauglichen Entwurf von Geobrowser-Inhalten sind vorhanden, und es gilt: Es gibt ein Usability-Werkzeug für Geobrowser. Für künftige Versionen müssen Schwächen des Systems kuriert werden und es wird empfohlen, **geolyzer** vorher einem langfristigen Praxistest zu unterziehen. Leider stellt ein Hauptproblem für künftige Versionen die fallen gelassene Unterstützung der API seitens Google dar.

Die Tatsache, dass seit gut einem Jahr kein Update der Google Earth COM API mehr veröffentlicht wird, ist besonders für das Projekt **geolyzer** ein großer Nachteil. Einige Wünsche für spätere Versionen lassen sich nicht realisieren, so z.B. das Tracking der Verlinkungen oder die Abbildung der tatsächlichen Kameraposition als Ergänzung zum Kamerafokus. Gerade hier wäre es wünschenswert, wenn die an einigen Stellen mangelhafte API weiterentwickelt, verbessert und nicht vernachlässigt würde.

Andererseits ist es auch verständlich, dass Google sich zu diesem Schritt entschlossen hat. Das Browser-Plugin, sofern es denn einmal die Kinderkrankheiten überstanden hat, wird bezüglich der Features gleichmächtig zum Client-System und ist für Nutzer im Browser schnell einsatzbereit. Hier muss nicht erst ein 15 MB großes Programm installiert werden, um Inhalte darzustellen, von denen eh viele mit Web-Inhalten verlinkt sind. Für Nutzer ist es angenehmer, Aktivitäten wie z.B. die Urlaubsplanung insgesamt im Webbrowser zu vollziehen und nicht zwischen Geo- und Webbrowser pendeln zu müssen.

Sollten also die Dienste des **geolyzer** auf dem Markt angenommen werden, so muss früher oder später die Funktionalität auf das Browser-Plugin portiert werden. Für Usability-Tests würden sich, zusätzlich zu den schon jetzt vorhandenen Möglichkeiten, neue Perspektiven eröffnen. Probanden müssten nicht einmal mehr vor Ort einen Nutzertest mit dem **geolyzer** durchführen, sondern könnten sich online anmelden und via Browser-Plugin den Test daheim durchführen – wobei alle Nutzer- und Aufnahmedaten über das Internet an einen Firmenserver gesendet werden. Das Rekrutieren von Probanden in ausreichender Anzahl würde selbst für Kleinunternehmen kein Problem mehr darstellen und der zeitliche Aufwand würde sich minimieren. Die Kosten verringern sich und besonders die auto-evaluierenden Funktionen, wo *»je mehr Probanden desto besser«* gilt, hätten bei einer großen Anzahl an Online-Probanden generell eine höhere Aussagekraft.

Die Funktionalität ist in C# bereits vorhanden, und es sollte im geeigneten Team maximal fünf Monate dauern, um dies in Javascript bzw. AJAX⁽⁷⁸⁾ zu *»verwandeln«*, so dass mithilfe der API des Browser-Plugins dieselbe Funktionalität vorhanden ist. Webprogrammierung unterscheidet sich deutlich von objektorientierter Programmierung, so dass der Akt mit der alleinigen Anpassung der Syntax bei weitem nicht vollzogen wäre.

Alternativ zum Browser-Plugin könnte die Funktionalität auch für andere Geobrowser wie z.B. NASA Worldwind umgesetzt werden, die ebenso die KML unterstützen und über eine öffentliche Programmierschnittstelle verfügen. Allerdings könnte sich der eher geringe Marktanteil im Vergleich zu Google Earth sowie die höhere Betonung als GIS-Werkzeug als hinderlich erweisen.

Als letzter Gedanke bietet sich an, dieses *»Kompakt-Usability-Labor«* **geolyzer** nicht nur auf Geobrowser-Inhalte zu beschränken, sondern Ähnliches unter anderem Namen auch in anderen Bereichen bzw. auf allgemeiner Produktebene zu wagen. Ein Novum wie die Auto-Evaluation könnte zweifelsfrei ebenso im Web-Bereich funktionieren – und generell gilt, dass nicht alle Unternehmen bereit sind, für ihre Produkte ein zeit- und kostenintensives Usability-Labor zu beauftragen. Ist man bereit, gewisse Abstriche in Kauf zu nehmen, so tut es vielleicht auch die Software-Alternative.

(78) AJAX – Asynchronous Javascript and XML

Anhang

A.1	Datenschema geolyzer	109
A.2	Klassendiagramm geolyzer	110
A.3	Screenshot geolyzer	111
A.4	Exemplarischer Probandenreport.	112
A.5	Exemplarischer Heuristik Report	113
A.6	Exemplarischer geoReport	114
A.7	Ergebnis <i>nuterra</i> Usability-Fragebogen.	115
A.8	Mängel und bekannte Probleme	117
A.9	Wünsche an künftige Versionen	118

A.1 Datenschema geolyzer

Die drei neuen Relationen, die im Rahmen dieser Arbeit hinzugekommen sind, sind mit einer hellen Farbe kenntlich gemacht.

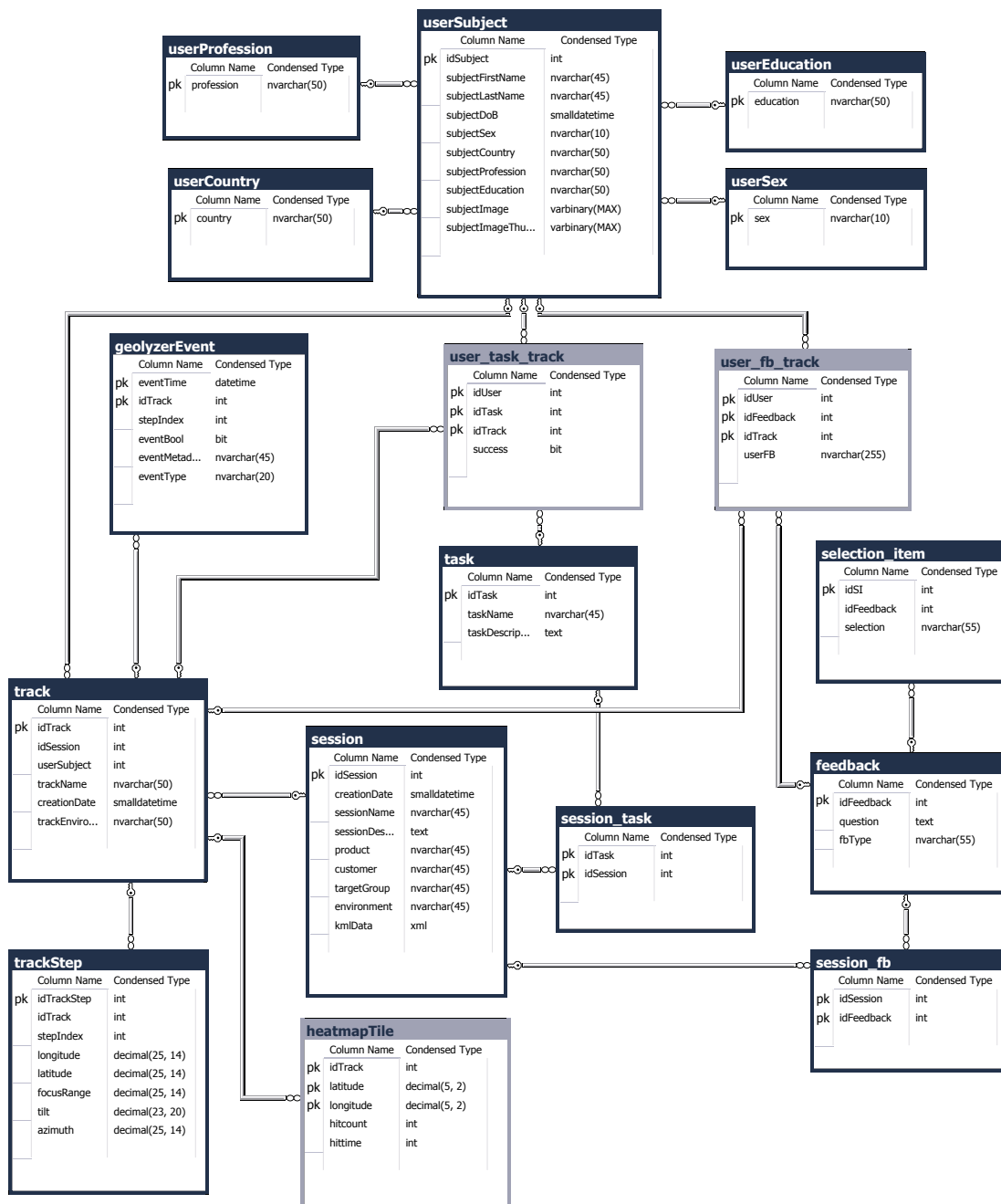


ABBILDUNG XXV:
Vollständiges Datenschema geolyzer

A.2 Klassendiagramm geolyzer

Alle Klassen, die während dieser Arbeit neu entstanden, sind über dunkle Farbe kenntlich gemacht. Das Klassendiagramm enthält als Auszug die wichtigsten Pakete und Assoziationen.

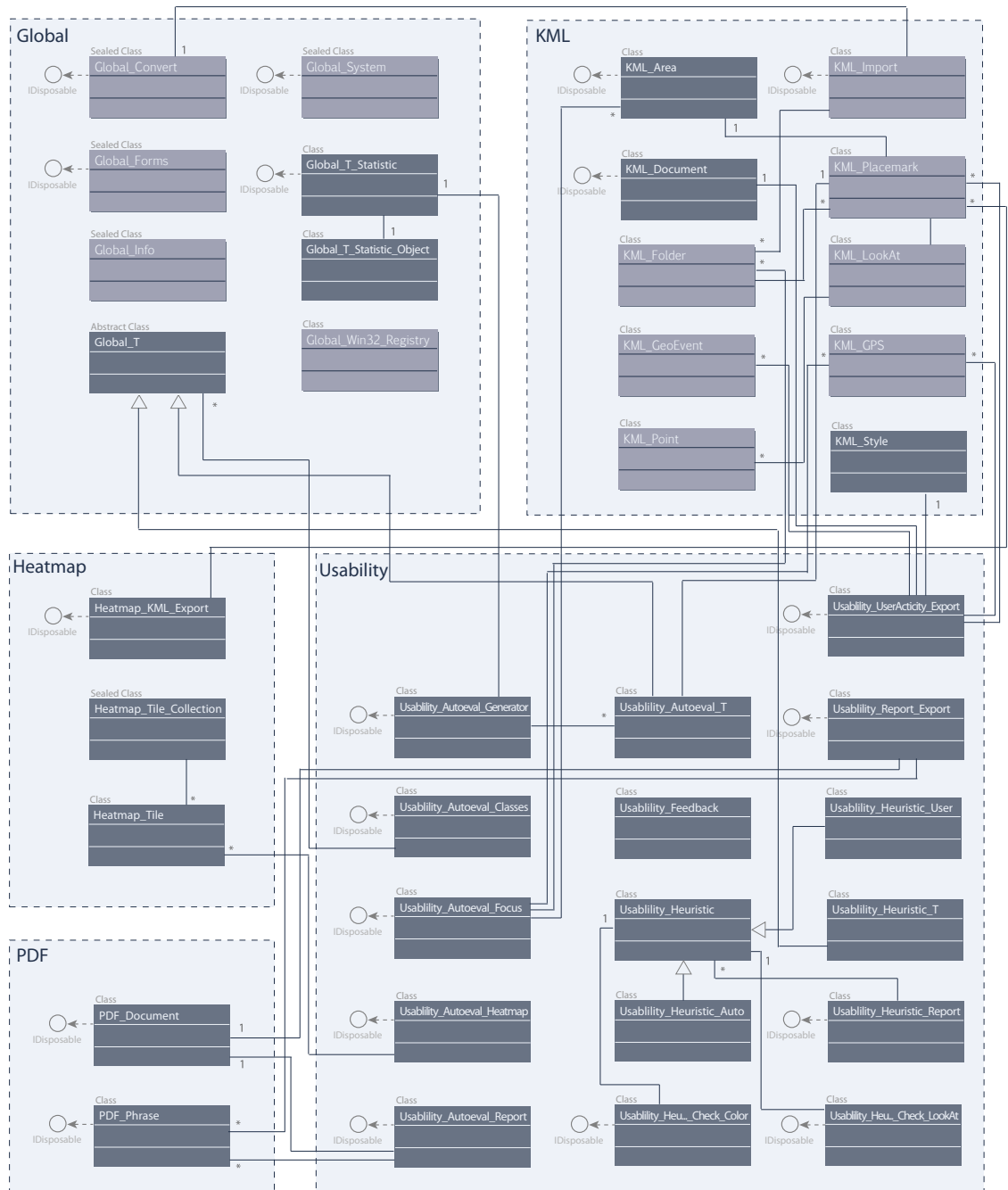


ABBILDUNG XXVI:
Klassendiagramm geolyzer (Auszug)

A.3 Screenshot geolyzer

Um sich ein Bild vom fertigen Produkt zu machen, soll ein Screenshot der laufenden Anwendung verdeutlichen, wie der **geolyzer** aussieht und aufgebaut ist. Der bekannte Google Earth Globus ist über ein Panel in die Anwendung integriert. Links befindet sich die Navigation ähnlich der von Google Earth. Unter dem Globusfenster dient der **geoPlayer** zum Aufnehmen und Wiedergeben einzelner Aufnahmen innerhalb einer Sitzung. Über das obige Menü sind alle Verwaltungs- und Exportfunktionen zu erreichen.

Der Screenshot wurde bei einer Bildschirmauflösung von 1200x800 erstellt. Die Software **geolyzer** benötigt eine Mindestbildschirmauflösung von 1024x768, bei höheren Auflösungen skalieren sich Steuerelemente automatisch.

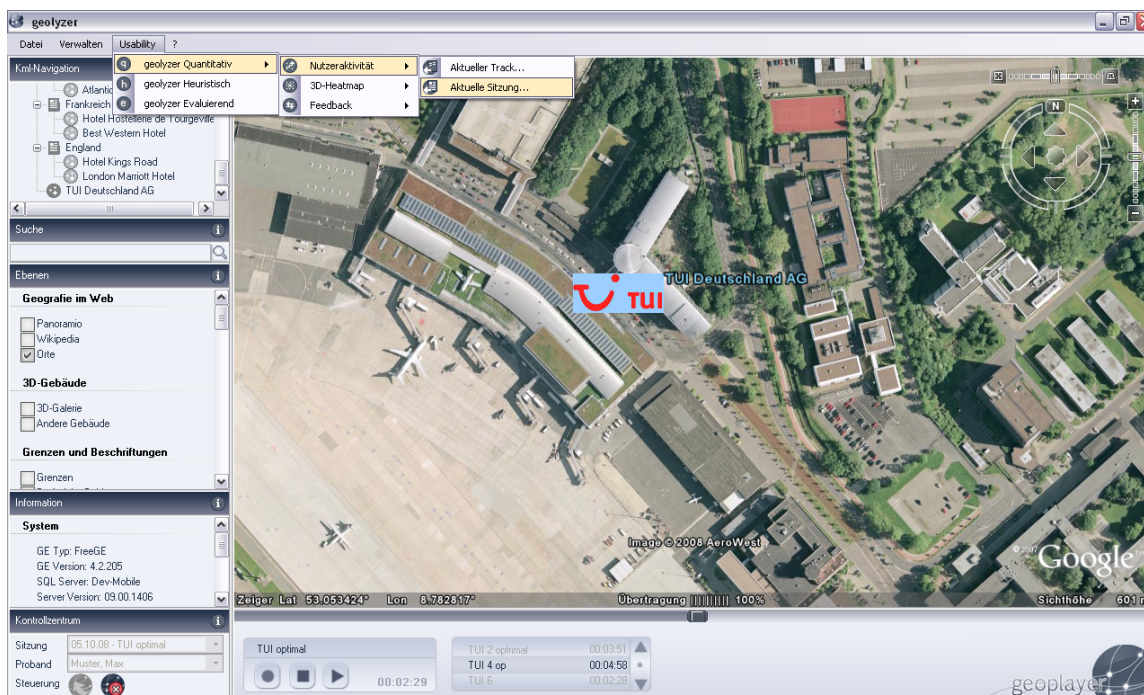


ABBILDUNG XXVII:
Screenshot **geolyzer**
Satellitenbilder ©AeroWest

A.4 Exemplarischer Probandenreport

Ein Auszug des *nuterra* Probandenreport, der im PDF-Format aus **geolyzer** zur Veranschaulichung der Probandenmeinungen exportiert wurde, ist abgebildet.


geolyzer

Probandenreport

26.09.2008 16:18:33

3.1. Personendaten

Proband: *Name nachträglich entfernt*
Geburtsdatum: 09.12.1966 00:00:00
Herkunft:
Bildungsgrad:
Beruf:

3.4. Feedback

Sitzung: Weltreise Ordner Nav
Aufnahmetitel: winter ordner nav
Datum: 23.09.2008 12:26:00
Frage: Wie gut sind Sie mit der Navigation zurecht gekommen?
Typ: evaluation
Antwort: *weniger gut*

3.2. Allgemeine Statistiken

An Sitzungen teilgenommen: 3
Aufnahmen durchgeführt: 3
Feedback abgegeben: 9
Längste Aufnahme: 00:08:35
Kürzeste Aufnahme: 00:01:58

3.3. Abgegebene Wertungen

Sehr gut: 2
Gut: 4
Neutral: 0
Schlecht: 0
Sehr schlecht: 0

3.4. Feedback

Sitzung: Weltreise ohne Nav
Aufnahmetitel: Winter ohne Nav
Datum: 23.09.2008 12:17:00
Frage: Hatten Sie das Gefühl, geführt zu werden?
Typ: evaluation
Antwort: *schlecht*


3.4. Feedback

Sitzung: Weltreise ohne Nav
Aufnahmetitel: Winter ohne Nav
Datum: 23.09.2008 12:17:00
Frage: Denken Sie, alle wesentlichen Infos gesehen zu haben?
Typ: selection
Antwort: *Ja*

ABBILDUNG XXVIII:
nuterra Probandenreport (Auszug)

A.5 Exemplarischer Heuristikreport

Ein Auszug aus dem Heuristikreport, der nach dem visuamundo Nutzertest generiert wurde, ist dargestellt.


geolyzer

Heuristiken Report

Erstellt: 06.10.2008 21:33:08

Heuristik: Überprüfung der Helligkeitsdifferenz zwischen Hintergrund- und Textfarben

Typ: automatisch

Anzahl an <BalloonStyle>-Tags: 16

Geeignete Farbkontraste: 1

<Balloon>-Tag Nr.: 1

Hintergrundfarbe: F02133

Textfarbe: D4C1AB

Kontrastverhältnis: 75

<Balloon>-Tag Nr.: 2

Hintergrundfarbe: FCB201

Textfarbe: FAC131

Kontrastverhältnis: 10

<Balloon>-Tag Nr.: 4

Hintergrundfarbe: FFFFFF

Textfarbe: FF0000

Kontrastverhältnis: 130

<Balloon>-Tag Nr.: 13

Hintergrundfarbe: F02133

Textfarbe: D4C1AB

Kontrastverhältnis: 75

Heuristik: War die visuelle Konsistenz der Inhalte gegeben?

Typ: Probandenfeedback

Durchschnittliche Zustimmung: 31%

Heuristik: Waren Inhalte so aufbereitet, dass sie sich nicht überdeckt haben?

Typ: Probandenfeedback

Durchschnittliche Zustimmung: 75%

Heuristik: Wurde eine klare und gut verständliche Sprache verwendet?

Typ: Probandenfeedback

Durchschnittliche Zustimmung: 62%

Heuristik: Wurden neue Inhalte aufgrund durchgängiger Struktur wiedererkannt?

Typ: Probandenfeedback

Durchschnittliche Zustimmung: 75%

Heuristik: Überprüfung aller Ortsmarken auf Existenz eines LookAt-Tags

Typ: automatisch

Anzahl an Ortsmarken: 16

Ortsmarken mit LookAt-Placemark: 13

ABBILDUNG XXIX:
visuamundo Heuristikreport (Auszug)

A.6 Exemplarischer geoReport

Auszüge der **wichtigsten Metriken** des **geoReport**, der nach dem visuamundo Nutzer-test exportiert wurde. Der vollständige Report umfasst 14 Seiten.


geolyzer

geoReport

nach Common Industry Format v1.1
Erstellt: 06.10.2008 21:31:34

Testleiter: Dirk Koester
Kunde: TUI AG

Firma: dirkoester.de
Telefon: 0176-77146234
E-Mail: info@dirkoester.de
Adresse: Calvinstr.39, 28217-Bremen

1. Sitzungsdaten

Name: TUI Reiseportal
Erstellt am: 05.10.2008 23:03:00
Sitzungsbeschreibung: Die TUI möchte ihr Angebot auf die Nutzung von Geobrowser-Inhalten verlagern.
Anzahl an Aufnahmen: 3
Anzahl an zugewiesenen Aufgaben: 2
Anzahl an zugewiesenem Feedback: 9

5.5 Evaluation Gesamtzeit

Anzahl an Ortsmarken: 16
Anzahl an Aufnahmen: 3

Klasse A
Ortsmarke: Sinkaputra Olioliranges
Höchste Gesamtzeit: 0,5 Sekunden
Geringste Gesamtzeit: 0 Sekunden
Durchschnittliche Gesamtzeit: 0,12 Sekunden
Median der Gesamtzeit: 0 Sekunden
Standardabweichung: 0,25 Sekunden
Aufnahmen ohne signifikante Abweichung: 3 / 4

5.3 Autoheuristiken

Überprüfung Helligkeitsdifferenz
Anzahl an <BalloonStyle>-Tags: 16
Geeignete Farbkontraste: 1

Vorhandensein von LookAt-Tags
Anzahl an Ortsmarken: 16
Ortsmarken mit LookAt-Tag: 13

5.6 Evaluation Inaktivität

Anzahl an Ortsmarken: 16
Anzahl an Aufnahmen: 4

Ortsmarke: Atlantico Golfe Hotel
Aufnahmen ohne signifikante Abweichung: 1 / 3

Aufnahme 1:
Inaktivität: 7 Sekunden
Erwartungswert: 0 Sekunden
Standardabweichung: 5,55 Sekunden
Abweichung vom Erwartungswert: 7 Sekunden

Aufnahme 2:
Inaktivität: 2,5 Sekunden
Erwartungswert: 0 Sekunden
Standardabweichung: 5,55 Sekunden
Abweichung vom Erwartungswert: 2,5 Sekunden

Aufnahme 3:
Inaktivität: 21 Sekunden
Erwartungswert: 0 Sekunden
Standardabweichung: 5,55 Sekunden
Abweichung vom Erwartungswert: 21 Sekunden

ABBILDUNG XXX:
visuamundo geoReport (Auszug)

A.7 Ergebnis nuterra Usability-Nutzerfragen

Getestet wurden fünf Probanden, denen im Anschluss einer Sitzung vordefinierte Fragen zur Beantwortung übergeben wurden. Im Folgenden ein Überblick über Fragen und Antwortverteilung.

1. Welche Demo war für Sie am ansprechendsten?

- » Ohne Navigtion [1/5]
- » Ordernavigation [1/5]
- » Geführte Navigation **[3/5]**

2. Welche Navigation war für Sie am intuitivsten?

- » Ohne Navigtion [1/5]
- » Ordernavigation [0/5]
- » Geführte Navigation **[4/5]**

3. Bei welcher Demo haben Sie das Gefühl, die meisten Informationen bekommen zu haben?

- » Ohne Navigtion [1/5]
- » Ordernavigation [1/5]
- » Geführte Navigation **[3/5]**

4. Wie wichtig ist Ihnen eine geführte Darstellung?

- » Sehr wichtig [0/5]
- » Eher wichtig [1/5]
- » Eher unwichtig **[4/5]**
- » Unwichtig [0/5]

5. Wie wichtig ist Ihnen eine ansprechende Umsetzung?

- » Sehr wichtig [1/5]
- » Eher wichtig **[3/5]**
- » Eher unwichtig [1/5]
- » Unwichtig [0/5]

6. Wie erfahren waren Sie mit Google Earth vor dem Test?

- » Unerfahren [1/5]
- » Wenig erfahren [1/5]
- » Mäßig erfahren [0/5]
- » Erfahren **[3/5]**
- » Experte [0/5]

7. Wie sind Sie mit der Handhabung des **geolyzer** zurechtgekommen?

- » Sehr gut [1/5]
- » Eher gut **[3/5]**
- » Eher schlecht [1/5]
- » Schlecht [0/5]

8. Wie ähnlich ist die Navigation mit dem Tool, verglichen mit Google Earth?

- » Identisch [0/5]
- » Sehr ähnlich [2/5]
- » Ähnlich **[3/5]**
- » Anders [0/5]
- » Völlig anders [0/5]

9. Ist Ihnen bewusst, dass man auf die Homepage der Weltreise kommen konnte?

- » Ja [1/5]
- » Nein **[4/5]**

10. Wo sollte Ihrer Meinung nach ein Klick auf das Titelbanner hinführen?

- » Homepage der Weltreise [4/5]
- » Start-Ortsmarke **[0/5]**
- » Streckenübersicht [0/5]
- » Sonstiges [1/5]

Anmerkung eines Probanden zu **10:**

- » *»Wirkte wie Werbung«*

A.8 Mängel und bekannte Probleme

Im Folgenden eine Auflistung an bekannten Problemen sowie noch vorhandenen Mängeln des **geolyzer**. Lösungsansätze für künftige Versionen werden genannt.

Defizit	Kurzbeschreibung	Lösung
<i>Scrollrad</i>	Das Zoomen in Google Earth via Scrollrad wird nicht unterstützt.	Workaround mit manuellem Zoom
<i>Nord-Ausrichtung</i>	Die automatische Nord-Ausrichtung durch die N-Taste ist nicht gegeben	Wird von der GE-API nicht unterstützt
<i>Visualisierung der Aufgaben während der Aufnahme</i>	Die Visualisierung der Aufgaben als Liste innerhalb der Informationen-Listbox während einer Aufnahme ist umständlich und nicht intuitiv	Neues Visualisierungskonzept
<i>Mehrere KMLs pro Sitzung</i>	Die Vorgabe, nur eine KML-Datei einer Sitzung zuweisen zu können, stößt auf Ablehnung, da Inhalte oft auf mehrere KML-Dateien verteilt sind	Ist machbar, aber auch aufwändig, da hoher Eingriff ins Programm
<i>Automatisches Öffnen der Ortsmarke beim Anfliegen</i>	Beim Anwählen einer Ortsmarke aus der GE-Ordnerstruktur wird diese nicht geöffnet	Bug in der GE-API. Nicht behebbar
<i>Transparenzen</i>	Transparenzeinstellungen für Inhalte fehlen.	Wird von der GE-API nicht unterstützt
<i>Export von einzelnen Aufnahmen nicht intuitiv</i>	Es ist nicht intuitiv, zum Export einer einzelnen Aufnahme diese anhand des geoPlayers auszuwählen. Optimaler wäre es, die Aufnahme im Dialogfeld zum Export angeben zu können.	Wird ab der nächsten Version umgesetzt
<i>Verbindung zum SQL-Server im Netzwerk</i>	Die Microsoft SQL-Server 2005 Express Version bereitet bei einigen Systemen Schwierigkeiten beim Remote-Zugriff. Bei der Standard- oder Developer-Version existiert dieses Problem nicht.	Eingehende Untersuchung des Problems und ggf. alternative SQL-Verbindungsvorgaben realisieren

TABELLE VIII: Mängel und bekannte Probleme

A.9 Wünsche an künftige Versionen

Durch den Praxistest wurde deutlich, welche Funktionalität in künftigen Versionen Sinn macht und die Arbeit mit den **geolyzer** verbessert.

Wunsch	Kurzbeschreibung	Machbarkeit
<i>Online-Version</i>	Neben der Client-Version wäre eine Online-Version auf Basis des Browser-Plugins wünschenswert, so dass Probanden daheim Usability-Tests durchführen können und die Daten an einen zentralen Server gesendet werden.	Portierung der geolyzer -Funktionalität auf das Browser-Plugin machbar. Geschätzter Aufwand: vier bis sechs Monate
<i>Farben für Probanden definieren</i>	Äußerst hilfreich wäre es, Probanden eine Farbe zuweisen zu können, so dass beim Tracking-Export einer Sitzung anhand der Farben sofort erkannt würde, welche Aufnahme zu welchem Probanden gehört.	Wird ab der nächsten Version umgesetzt
<i>Tracking für Verlinkungen innerhalb Ortsmarken</i>	Neben dem Tracking zu Nutzerereignissen, Kamerafokus und den Mauspositionen inklusive der Verweildauer wäre es hilfreich, wenn auch aufgezeichnet würde, welche Links in den Ortsmarken angeklickt werden.	Wird von der GE-API nicht unterstützt. Theoretisch mit dem Browser-Plugin möglich
<i>Zeit für einzelne Aufgaben messbar</i>	Es sollte im Programm möglich sein, während einer Aufnahme den Beginn und den Abschluss einer Probanden-Aufgabe zu kennzeichnen und das Zeitintervall zu messen.	Wird ab der nächsten Version umgesetzt
<i>Gesamtzeit für Ortsmarken definieren</i>	Für alle Ortsmarken soll ein Erwartungswert für die Gesamtzeit definiert werden, wie lange Probanden den entsprechenden Inhalt fokussieren sollten. Nach der Aufnahme kann somit die tatsächliche Zeit mit dem Erwartungswert verglichen werden.	Wird ab der nächsten Version umgesetzt
<i>Ortsmarken für Evaluation deaktivieren</i>	Ortsmarken, die nicht direkt zum Inhalt gehören, z.B. Flughäfen in der Nähe, sollten für die Bewertung deaktiviert werden können.	Wird ab der nächsten Version umgesetzt

TABELLE IX: Wünsche an künftige Versionen

Quellen

[COMMONGIS]

Löhr, S.

„Thematische 3D-Kartographie unter Verwendung von CommonGIS und Google Earth“

http://www.iais.fraunhofer.de/fileadmin/images/pics/Abteilungen/SPADE/Diplomarbeit-Stefa_Loehr.pdf

[FIT-USABILITY]

Fraunhofer Institut

„Willkommen im Usability Begriffszoo“

<http://www.fit-fuer-usability.de/1x1/basics/begriffszoo.html>

Stand: **02.08.2008**

[FRIEDMAN 2007]

Friedman, V.

„Praxisbuch Web 2.0“

Bonn: Galileo Press, 2007

[GEOWEB]

Internet3D

“Geoweb and Geobrowsers prepare 3D Internet“

<http://www.internet3d.org/>

Stand: **16.08.2008**

[GE THEMA]

Bleisch, S.

„Google Earth, NASA World Wind und Co. – Wenn Geoinformation zum Modethema wird“

http://www.geomatik.ch/fileadmin/download/2006/Fach/FA_2_2006_1.pdf

[GUIDELINES 2006]

Shneiderman, B.

“Research-Based Web Design & Usability Guidelines”

U.S. Government Printing Office, Washington, DC, 2006

ISBN: 0-16-076270-7

[KML]

Google

“Google Earth KML Tutorial”

<http://earth.google.de/kml/index.html>

Stand: **11.08.2008**

[KRUG 2002]

Krug, S.

„Web Usability – Don’t make me think!”

Bonn, mitp-Verlag, 2002

ISBN: 3-8266-0890-9

[MACDONALD 2006]

MacDonald, M.

“Pro .NET 2.0 Windows Forms and Custom Controls in C#”

New York: Apress, 2006

ISBN: 1-59059-439-8

[MERTINS 2006]

Mertins, D., Neumann, J.

„Programmierhandbuch SQL Server 2005“

Bonn: Galileo Press, 2006

ISBN: 3-89842-684-X

[MSDN LIBRARY]

Microsoft Developer Network

<http://msdn.microsoft.com/de-de/default.aspx>

[NBD-TECH]

Calculating the Perceived Brightness of a Color

<http://www.nbdtech.com/blog/archive/2008/04/27/Calculating-the-Perceived-Brightness-of-a-Color.aspx>

Stand: **06.09.2008**

[NIELSEN]

Nielsen, J.

"Usability Einmaleins"

<http://www.usability.ch/Alertbox/Main.htm>

Stand: **05.08.2008**

[PROCONTEXT]

Geis, T.

„Neues aus der Usability-Normung“

ProContext GmbH

<http://www.procontext.com/de>

Stand: **05.08.2008**

[REGIS]

Schulte, B.

„KML als Instrument zur Visualisierung historischer Geodaten in Google Earth“

http://benneten.de/referate/2007-07-26-regis-ge_tu/referat_regis-ge_tu_benneten.de.pdf

[RLOEWY]

Koester, D.

„Raymond Loewy – Pionier Amerikanischen Industriedesigns“

http://www.dirkoester.de/content/text/raymond_loewy.pdf

[SCHARL 2007]

Scharl, A., Tochtermann, K.

“The Geospatial Web – How Geobrowsers, Social Software and the Web 2.0 are Shaping the Network Society”

London: Springer Verlag, 2007

ISBN: 1-84628-826-6

[STAPELKAMP 2007]

Stapelkamp, T.

„Screen- und Interfacedesign, Gestaltung und Usability für Hard- und Software“

Berlin: Springer Verlag, 2007

ISBN: 978-3-540-32949-7

[WIKI GEOREFERENZIERUNG]

Wikipedia

<http://de.wikipedia.org/wiki/Georeferenzierung>

Stand: **29.07.2008**

[WIKI KML]

Wikipedia

http://de.wikipedia.org/wiki/Keyhole_Markup_Language

Stand: **12.08.2008**

[WIKI NASA]

Wikipedia

http://de.wikipedia.org/wiki/NASA_World_Wind

Stand: **03.08.2008**

[WIKI GE]

Wikipedia

http://de.wikipedia.org/wiki/Google_Earth

Stand: **09.08.2008**

Abbildungen

Abbildung I:	Kommunikationswege Earth Viewer	18
Abbildung II:	Rotationsellipsoid als vereinfachtes Modell der Erde	23
Abbildung III:	Kameraposition und Manipulation in KML	27
Abbildung IV:	KML Geometrieobjekte LineString sowie Polygon	31
Abbildung V:	KML-Inhalte negativ und positiv aufbereitet	34
Abbildung VI:	Anteil Behinderungen USA 2003	42
Abbildung VII:	Prinzipieller Aufbau einer Heatmap.	50
Abbildung VIII:	Historie geolyzer.	54
Abbildung IX:	Anwendungsfalldiagramm Ist-Zustand	55
Abbildung X:	Architektur ADO.NET.	57
Abbildung XI:	Windows Control Hierarchie.	60
Abbildung XII:	geolyzer – zweischichtige Architektur Client/Server.	62
Abbildung XIII:	geolyzer – Paketdiagramm	63
Abbildung XIV:	Skizze zur Abbildung der Nutzeraktivität	67
Abbildung XV:	Zwei Konzepte zur Visualisierung von Heatmaps	70
Abbildung XVI:	Klasse KML_Document	82
Abbildung XVII:	Nah aneinander liegende Koordinaten filtern.	83
Abbildung XVIII:	Klasse Heatmap_Tile_Collection und Heatmap_Tile	85
Abbildung XIX:	Exportierte 3D-Heatmap, Nutzeraktivität	86
Abbildung XX:	Durchschnittliche Nutzerwertung	87
Abbildung XXI:	Klassen zu Usability Heuristiken.	88
Abbildung XXII:	Klassen zu Usability Statistiken	90
Abbildung XXIII:	Klassen zum geoReport	91
Abbildung XXIV:	visuamundo Heatmap und Nutzer-Ereignisse	97
Abbildung XXV:	Datenschema geolyzer.	109
Abbildung XXVI:	Klassendiagramm geolyzer	110
Abbildung XXVII:	Screenshot geolyzer.	111
Abbildung XXVIII:	Exemplarischer Probandenreport	112
Abbildung XXIX:	Exemplarischer Heuristikreport	113
Abbildung XXX:	Exemplarischer geoReport	114

Tabellen

Tabelle I:	Herkunft und Auflösung Google Earth Bildmaterial	19
Tabelle II:	Navigationsmöglichkeiten im Webbrowser	36
Tabelle III:	Navigationsmöglichkeiten im Geobrowser	36
Tabelle IV:	Soll-Funktionalität geolyzer	64
Tabelle V:	Konzept geolyzer Heuristisch	72
Tabelle VI:	Report auto-evaluierend gemäß CIF	76
Tabelle VII:	Index als Bewertung der Inhalte.	81
Tabelle VIII:	Mängel und bekannte Probleme.	117
Tabelle IX:	Wünsche an künftige Versionen	118

Abkürzungen

ADO	ActiveX Data Objects
ANSI	American National Standards Institute
API	Application Programming Interface
CAD	Computer Aided Design
CIF	Common Industry Format
DCM	Data Control Language
DDL	Data Definition Language
DML	Data Manipulation Language
GDI	Graphics Device Interface
GIS	Geoinformationssystem
GML	Geography Markup Language
ISO	International Organization for Standardization
KML	Keyhole Markup Language
NIST	National Institute of Standards and Technology
ODBC	Open Database Connectivity
OGC	Open Geospatial Consortium
PDF	Portable Document Format
RSS	Really Simple Syndication
SQL	Structured Query Language
UDF	User Defined Functions
W3C	World Wide Web Consortium
XML	Extensible Markup Language

Index

A

ADO.NET 6, 53, 54, 56, 57, 123, 128
Adresskodierung 21
American National Standards Institute 59, 125
Application Programming Interface 16, 125
ArcGIS Explorer 17, 20, 128
Assist Rate 77
Auto-Evaluierend 6, 7, 75, 102, 103

B

Breitengrad 24, 30, 100
Browser Plugin 19, 128

C

CDATA 27
Clicktracks Appetizer 51
Common Industry Format 49, 125, 128
Completion Rate 77, 81
Computer Aided Design 15, 125
coordinates-Tag 30
Crowdsourcing 15
Custom Control 60

D

DataSet 57, 58
DataTable 57, 58, 68, 92, 93
Derived Control 60, 61, 128
description-Tag 27
DMS-Format 24
Document-Tag 28

E

Efficiency Product Level 77
Error Rate 77
extrude-Tag 30
Eye-Tracking 48, 50, 51, 102

F

Fat-Client 62, 128
Flickrvision 16
Folder-Tag 29

G

GDI+ 61, 71, 87, 128
Geocoding 22
Geoextraktion 5, 22, 128
Geography Markup Language 25
Geoid 23
Geoinformationssystem 15, 22, 125, 128
Geokodierung 5, 20, 21, 22, 26, 128
Geoparser 22
Georeferenzierung 21, 122
Geotagging 21, 128
Geozentrische Breite 24
Google Analytics 51

H

Hauptbreitenkreis 24
heading-Tag 27
Heuristische Verfahren 47

K

KMZ-Dateiformat 25

L

labelStyle-Tag 27
 Längengrad 22, 24, 30
 Link-Tag 32
 LookAt-Tag 27, 72

M

Mean Task Time 77
 Microsoft SQL Server 56, 62
 Microsoft Visual Studio 56

N

NASA World Wind 14, 17, 20, 25, 33, 119, 128
 NetworkLink 32
 Nullmeridian 24
 nuterra 6, 7, 9, 35, 74, 94, 95, 99, 100, 108, 115

O

Open Database Connectivity 57, 125
 Open Geospatial Consortium 25, 125
 Ortsabhängige Dienste 17
 Owner-Drawn Control 60, 128

P

Phantomverläufe 100, 102
 Placeopedia 16
 Point-Tag 27

Q

Qualitative Verfahren 47, 51

R

Referenzellipsoid 23
 refreshInterval-Tag 32
 REGIS:GE 16
 Rotationsellipsoid 23, 123

S

Satisfaction Rate 77, 81, 128
 Screen-Capturing 47, 48, 51
 SketchUp 15, 25
 Standardabweichung 79, 80, 91, 93, 128
 Structured Query Language 59, 125
 styleurl-Tag 28

T

Tracking-Verfahren 47, 51, 103
 Transact-SQL 6, 53, 59, 128

U

User Control 60, 61
 using-Direktive 58

V

visuamundo GbR 6, 9, 35, 65, 74, 94

W

Web 3.0 17
 WGS84 23, 128
 Wireless Communication Lab 16
 World Wide Web Consortium 42, 125

Google Earth
 Keyhole Markup Language
 Usability Geoinformationssystem
 AnalyseHeuristikUserFeedback EarthViewerOwner-DrawnControlGeoextraktionGeokodierung.NETFramework
 Geodätisches DatumGML Transact-SQL 3D-Heatmap NutzeraktivitätGeoreferenziellesWeb CommonIndustryFormatNASAWorldWindSQL
 Server 2005 Custom Controlsgeolyzer GoogleEarthCOMAPI BrowserPluginArcGISExplorer
 geoReportWeb3.0 GeotaggingWGS84LatitudeLongitude XMLPlacemarkWebbrowsersvs.Geobrowser
 AccessibilityTrackingADO.NET 2.0 User Controls GDI+ Derived Control Fat-Client geoPlayer Auto-Evaluation
 Heatmap-Tile Standardabweichung Median Satisfaction Rate ABGR Phantomverlauf Kamerafokus Google Earth
 Keyhole Markup Language Usability Geoinformationssystem Analyse Heuristik User Feedback Earth Viewer
 Owner-Drawn Control Geoextraktion Geokodierung .NET Framework Geodätisches Datum GML Transact-SQL 3D-
 Heatmap Nutzeraktivität Georeferenzielles Web Common Industry Format NASA World Wind sql
 Server 2005 Custom Controls geolyzer Google Earth COM API Browser Plugin ArcGIS Explorer geoReport
 Web 3.0 Geotagging WGS84 Latitude Longitude XML Placemark Webbrowser vs. Geobrowser Accessibility Tra-
 cking ADO.NET 2.0 User Controls GDI+ Derived Control Fat-Client geoPlayer Auto-Evaluation Heatmap-Tile
 Standardabweichung Median Satisfaction Rate ABGR Phantomverlauf Kamerafokus Google Earth
 Keyhole Markup Language Usability Geoinformationssystem Analyse Heuristik User
 Feedback Earth Viewer Owner-Drawn Control Geoextraktion Geokodierung .NET
 Framework Geodätisches Datum GML Transact-SQL 3D-Heatmap Nut-
 zeraktivitätGeoreferenziellesWeb CommonIndustryFormatNASAWorld
 Wind SQL Server 2005 Custom Controls geolyzer Google Earth
 COM API Browser Plugin ArcGIS Explorer geoReport Web
 3.0 Geotagging wgs84 Latitude Longitude
 XML Placemark Webbrowser vs. Geobrow-
 ser Accessibility ABGR Phantom-
 verlauf Kamerafokus