Usability-Evaluation
im Rahmen von INVISIP

Bachelorarbeit

im Fach Information Engineering

an der

Universität Konstanz

vorgelegt von

Hans-Christian Jetter, Juni 2003

Zusammenfassung

Inhalt dieser Bachelorarbeit ist die Darstellung und Diskussion der verwendeten Methoden der Usability-Evaluation im Rahmen des INVISIP Projekts.

Neben einer kurzen Darstellung der Evaluationsergebnisse wird der Schwerpunkt vor allem auf die Erläuterung verschiedener Methoden der Usability-Evaluation gelegt und deren Anwendung innerhalb des Projekts diskutiert. Die Entscheidungen bezüglich der unterschiedlichen Varianten sowie deren individuelle Gestaltung und Durchführung werden ausführlich dargestellt.

Mit der Vorstellung des webbasierten Usability Tests und des universell einsetzbaren Usability-Frameworks werden die Erkenntnisse aus der Praxis der Evaluation im Entwurf webbasierter Methoden umgesetzt, die die Produktentwicklung und die post-deployment Phase bei zukünftigen Projekten unterstützen können.

In der Zusammenfassung werden abschließend alle behandelten Evaluations-Methoden gegenübergestellt und ein Leitfaden zur ökonomisch sinnvollen und ergänzenden Anwendung verschiedener Methoden der Usability-Evaluation entwickelt.

Abstract

This bachelor thesis presents and discusses the different techniques of usability evaluation applied in the INVISIP project.

Apart from a brief presentation of the evaluation results this thesis focusses on the explanation of the individual evaluation techniques and the discussion of their applicability to the project.

With the introduction of remote usability testing techniques and a usability software-framework the experiences from the INVISIP project are transferred into concepts for new remote evaluation tools, which can be used to support the development and the post-deployment phase in other projects as well.

Finally all discussed techniques are briefly recapitulated to give a decision-aid for economic and complementary usage of usability evaluation techniques for the future.

Inhaltsverzeichnis

1 Einleitung 6

2 Der INVISIP Metadaten-Browser 8

2.1 Die Visualisierungen des Metadaten-Browsers 10

2.2 Das LevelTable Design 11

2.3 Das GranularityTable Design 12

2.4 Der Scatterplot 14

3 Evaluation der INVISIP Mock-Ups 15

3.1 Kernfragen an die Evaluation der Mock-Ups 16

3.2 Anforderungen an die Mock-Ups 17

3.3 Gestaltung der Mock-Ups 18

3.4 Befragung der Zielgruppe 20

3.5 Laborevaluation der Mock-Ups 22

3.5.1 Eigenschaften des Usability-Tests im Labor 22

3.5.2 Durchführung der Laborevaluation bei INVISIP 23

3.5.3 Testmoderation 24

3.5.4 Auswertung 25

3.6 Webbasierte Evaluation der INVISIP Mock-Ups 27

3.6.1 Eigenschaften der webbasierten Usability-Evaluation (WUE) 27

3.6.2 Varianten der WUE für INVISIP 28

3.6.3 Variante 1: Weblogging mit WebQuilt 28

3.6.4 Variante 2: Usability-Webbefragung 30

3.6.5 Durchführung der webbasierten Evaluation 32

3.6.6 Beispiel: Ermitteln des Verständnisses für die GranularityTable 34
3.7 Zusammenfassung der Ergebnisse der Mock-Up Evaluationen 36
3.7.1 Benutzerprofil 37
3.7.2 Vergleich LevelTable vs GranularityTable 37
3.7.3 Testergebnisse für Scatterplot 40
4 Heuristische Evaluation der INVISIP Java-Implementation 41
4.1 Hintergrund und Durchführung der Evaluation 41
4.2 Zusammenfassung und Redesign-Vorschläge 42
5 Zukünftige Möglichkeiten der Evaluation für INVISIP 44
5.1 Evaluation im weiteren Projektverlauf 44
5.2 Der webbasierte Usability-Test 45
5.3 Das Usability-Framework für INVISIP 47
5.3.1 Grundprobleme des Usability-Frameworks 50
5.3.2 Architektur des Usability-Frameworks für Java 52
5.3.3 Quality Feedback und weitere Dienste im Usability-Framework 53
5.3.4 Datenformate und Visualisierungen für das Usability-Framework 54
5.3.5 Intelligente Software-Agenten für das Usability-Framework 54
6 Zusammenfassung 57
6.1 Tabellarische Gegenüberstellung der Evaluations-Methoden 58
7 Anhang 60
7.1 Anhang A: Laborevaluation 60
7.1.1 Aufgabenstellung LevelTable 60
7.1.2 Aufgabenstellung GranularityTable 63

7.1.3 Beispielauswertung einer Testsitzung ... 66

7.2 Anhang B: Webbasierte Evaluation ... 67

7.3 Anhang C: Heuristische Evaluation ... 68

7.3.1 Visibility of System Status ... 68

7.3.2 Consistency and Standards ... 70

8 Literaturverzeichnis ... 72

8.1 Veröffentlichungen zu GIS und GIS-Infrastruktur ... 72

8.2 Veröffentlichungen zu INVISIP/INSYDER ... 72

8.3 Veröffentlichungen zu visuellen Suchsystemen und zum Document Retrieval ... 73

8.4 Veröffentlichungen zur heuristischen Evaluation ... 73

8.5 Veröffentlichungen zu Usability-Questionnaires und Webbefragung ... 73

8.6 Veröffentlichungen zu Methoden & Techniken der Usability-Evaluation ... 73

Abbildungsverzeichnis

Abbildung 2-1: LevelTable (Level 2) 11

Abbildung 2-2: LevelTable (Level 1) 11

Abbildung 2-3: LevelTable (Level 3) 11

Abbildung 2-4: LevelTable (Level 4) 11

Abbildung 2-5: GranularityTable (kleine Gran.) 13

Abbildung 2-6: gemischte Granularity 13

Abbildung 2-7: GranularityTable (mittlere Gran.) 13

Abbildung 2-8: GranularityTable (max. Gran.) 13

Abbildung 2-9: Interaktion mit Scatterplot 14

Abbildung 2-10: LevelTable + Scatterplot 14

Abbildung 3-1: Verständnisfaktoren (-8 bis 8) für SuperTable 26

Abbildung 3-2: Schematische Darstellung der WUE 27

Abbildung 3-3: Normale Logfiles vs. Webquilt Logfiles 29

Abbildung 3-4: Screenshot aus der Webbefragung 34

Abbildung 5-1: Schematische Darstellung Remote Computing 45

1 Einleitung

Im Rahmen des EU-Projekts INVISIP (Information Visualization for Site Planning, IST 2000-29640) arbeiten verschiedene Projektpartner wie z.B. Forschungseinrichtungen und Unternehmen im IT-Bereich und der Bau-/Verkehrsplanung zusammen mit kommunalen Verwaltungen an der Entwicklung und Integration von visuellen Werkzeugen zur Unterstützung von Standortentscheidungen mithilfe von Geoinformationssystemen (GIS) [2:INVISIP]. Betrachtet man die Vielzahl anderer Projekte in diesem Bereich (beispielsweise die Initiativen der Bundesländer Brandenburg „GeoBroker“ am Fraunhofer-ISST oder Hessen „InGeoForum“ am Fraunhofer-IGD) wird deutlich, welche Relevanz der Entwicklung von Informationsinfrastruktur in diesem Bereich beigemessen wird [1:GeoBroker], [1:InGeoForum]. Handlungsbedarf besteht insbesondere wegen des hohen Grades von Intransparenz des Informationsmarkts für Geodaten, da dieser von unterschiedlichsten Standards und heterogenen Datenquellen durchzogen ist. Es gibt daher bislang kaum standardisierte Vorgehensweisen oder Schnittstellen zum Zugriff auf entsprechende Daten [1:MICUS].

Im Rahmen von INVISIP beschäftigt sich die AG Mensch-Computer Interaktion der Universität Konstanz insbesondere mit der Entwicklung eines Metadaten-Browsers, der als Document Retrieval Komponente die visuelle Suche nach raumbezogenen Daten und Dokumenten auf verschiedensten Datenbanken ermöglicht und den Benutzer bei der Anforderungen von planungsrelevanten Informationen in allen Phasen des Planungsprozesses begleitet und unterstützt [2:Göbel et al. 1].

Der aus den Ergebnissen des EU-Projekts INSYDER (Internet Systéme De Recherche, IST No. 29232 [2:INSYDER Web], [2:Mußler et al.]) heraus weiterentwickelte Browser stellt im Falle von INVISIP eines der zentralen Userinterfaces für die Benutzer aus der Anwendungsdomäne dar (Stadtplaner, Bauingenieure, Architekten, Planungsbüros etc). Aus diesem Grund ist eine usability-konforme Gestaltung des Browsers ein kritischer Faktor für den Gesamterfolg des Projekts.

Die Konformität des Metadaten-Browsers mit den Erkenntnissen des Usability-Engineerings wurde und wird daher durch User-Tests, User-Befragungen und durch die Anwendung von Heuristiken im Rahmen verschiedener Usability-Evaluationen gewährleistet, die die Entwicklung des Metadaten-Browsers von Anfang an begleiteten.

Im Folgenden werden die verschiedenen Varianten der zum Einsatz gekommenen Usability-Evaluationen erläutert und deren Anwendbarkeit bezogen auf die jeweiligen Projektphasen diskutiert. Die Entscheidungsfindung für die Evaluationstechniken sowie deren individuelle Gestaltung und Durchführung sollen dabei im Mittelpunkt stehen.

Da eine Diskussion der Evaluationsmethoden ohne Kenntnisse des zu evaluierenden Systems einen zu hypothetischen und unanschaulichen Charakter hat, wird im Kapitel 2 zunächst das Konzept des Metadaten-Browsers und der Entwicklungsstand zu Beginn des Projektpraktikums dargestellt.

In Kapitel 3 wird auf die Umsetzung dieses Entwicklungsstandes durch geeignete Mock-Ups, sowie auf deren Evaluation eingegangen. Dabei werden neben den verwendeten Techniken auch die Evaluationsergebnisse dargestellt, um den Nutzen für die Projektpraxis anschaulich wiederzugeben.

Kapitel 4 beschäftigt sich mit der heuristischen Evaluation der Java-Implementation der Mock-Ups. Auch hier wird sowohl auf methodische Aspekte wie auch auf konkrete Erkenntnisse eingegangen.

Schließlich wird in Kapitel 5 ein Ausblick auf zukünftige Methoden der Evaluation präsentiert, wobei insbesondere das Konzept eines Usability-Frameworks für Java-Applikationen entworfen wird, das die Evaluation und die post-deployment Phase von INVISIP oder anderen Projekten unterstützen könnte.

Kapitel 6 fasst die Ergebnisse aus den vorigen Kapiteln zusammen und stellt sie in Form einer kompakten Gegenüberstellung der behandelten Evaluations-Methoden dar. Somit ist sie als Leitfaden für eine ökonomisch sinnvolle und ergänzende Anwendung verschiedener Evaluationstechniken in zukünftigen Projekten zu verstehen.

2 Der INVISIP Metadaten-Browser

Auch wenn das INVISIP Projekt ursprünglich im Kontext von Geodaten angesiedelt ist, ist die Konzeption des Metadaten-Browsers universell einsetzbar und unabhängig von der Art der Datenbasis, die zukünftig beispielsweise auch medizinische oder wirtschaftliche Daten enthalten könnte. Auch die Integration in eine Websuchmaschine wäre ein denkbares Anwendungsgebiet.

Die Aufgabe des Metadaten-Browsers ist dabei die Bereitstellung eines grafischen Userinterfaces zur Durchführung von Suchprozessen auf Dokumenten-Datenbanken, das es erlaubt, die erhaltene Treffermenge mithilfe verschiedenster Visualisierungstechniken darzustellen, um damit dem Benutzer die Möglichkeit zu geben, die Relevanz der einzelnen Treffer für seine Fragestellung auch visuell beurteilen zu können [2:Klein et al.].

Dabei zieht der Metadaten-Browser die Metadatensätze zu den einzelnen Treffern - im Falle von Geodaten also z.B. Größe, Preis, Erscheinungsjahr, Maßstab von raumbezogenen Dokumenten wie Presseartikeln, Karten, Statistiken etc. - und die Relevanz ihres Inhalts zu den eingegebenen Suchbegriffen zur grafischen Darstellung und zur visuellen Bewertung durch den Benutzer heran [2:Göbel et al. 2].

Diese Möglichkeit zur visuellen Bewertung erlaubt dem Benutzer das schnelle und gezielte Isolieren und Auffinden von hilfreichen Dokumenten innerhalb großer Treffermengen und führt somit zu höherer Effizienz und Effektivität bei der Suche nach Informationen, da nicht nur eine Beschleunigung, sondern auch eine Präzisierung des Suchprozesses ermöglicht wird.

Mithilfe der Visualisierungen von INVISIP sind weitere wertvolle Charakteristika der Treffermenge erkennbar, beispielsweise die Rolle, die ein einzelner Suchbegriff innerhalb der Treffermenge für den gesamten Suchausdruck spielt, oder die grafische Darstellung der Metadaten der gefundenen Dokumente als Scatterplot (z.B. das Auftragen der Relevanz gegen das Erstellungsdatum). Diese Form der Visualisierung erlaubt es, Klassen oder Gruppen von ähnlich gearteten Dokumenten innerhalb der Treffermenge auf einen Blick identifizieren zu können (siehe 2.4).

Weiterhin soll dem Benutzer des Metadaten-Browsers ermöglicht werden, die Suchparameter (z.B. Suchbegriffe, Filtereigenschaften) über Interaktion mit den Visualisierungen dynamisch zu beeinflussen, d.h. es soll die Möglichkeit zu Dynamic Queries und zu einem visuellen Query Refinement geschaffen werden.

Dies erlaubt dem Benutzer die Abkehr von der traditionellen, iterativen Entwicklung von Suchausdrücken im formalen Dialog mit dem Suchsystem. Anstatt sich mit dem schrittweisen Hinzufügen oder Entfernen von Suchbegriffen je nach erhaltener Treffermenge zu beschäftigen, kann sich der Benutzer nun durch die Interaktion mit den Visualisierungen der gewünschten Treffermenge nähern, ohne dabei komplexe Suchausdrücke mit einer Vielzahl von Suchbegriffen oder booleschen Operatoren entwickeln zu müssen.

Durch die Visualisierungen lassen sich auch umfangreiche Treffermengen bei unpräzisen Anfragen deutlich leichter bewerten und explorieren. Dies hat insbesondere für die Benutzer einen hohen Wert, die in der Regel keine komplexen Suchausdrücke zur Eingrenzung entwickeln, sondern nur kleine und einfache Suchausdrücke verwenden, um anschließend die großen Treffermengen durch Browsing und Überprüfung der einzelnen Dokumente abzuarbeiten und nicht analytisch einzugrenzen. Diese browsing-orientierte Vorgehensweise ist häufig bei Gelegenheitsbenutzern und wenig erfahrenen Nutzern zu beobachten, da dort keine Kenntnisse über boolesche Ausdrücke und die grundlegenden Mechanismen von Suchmaschinen vorhanden sind. Mit zunehmender Geschwindigkeit des Internets und den damit immer kürzer werdenden Antwortzeiten hat sich diese Suchstrategie im Alltag als zunehmend effizient erwiesen, insbesondere dann, wenn der Benutzer dem Erlernen von analytischeren Techniken skeptisch gegenüber steht oder diesen Aufwand als nicht lohnenswert betrachtet [3:Marchionini]. Dabei unterstützt gerade auch der fliessende Übergang zwischen Metadaten- und Content-Ebene (siehe GranularityTable 2.3) den schnellen Zugriff auf die zu sichtenden Dokumente.

Ein weiterer Vorteil des Metadaten-Browsers gegenüber herkömmlichen Suchfunktionen ist die Wahlfreiheit des Benutzer in der Verwendung der individuellen Darstellungen. Im Gegensatz zum harten Wechsel zwischen dem normalem Angebot von Suchfunktionalität und der „Power-Suche“, wie z.B. bei Google o.ä. Internet-Suchmaschinen, können im Metadaten-Browser mit wachsender Erfahrung immer mehr Suchwerkzeuge niedrigschwellig hinzugezogen werden. Es werden somit also sowohl die erfahrenen analytisch arbeitenden Benutzer wie auch die browsing-orientierten Gelegenheitsnutzer unterstützt und zum Einsatz neuer Techniken angeregt.

2.1 Die Visualisierungen des Metadaten-Browsers

Der grundlegende Entwurf des INVISIP Metadaten-Browsers basiert auf den Ergebnissen der Evaluation und des Redesigns des Business-Intelligence Systems INSYDER und wurde bereits in verschiedenen Veröffentlichungen thematisiert ([2:Eibl et al.], [2:Göbel et al. 1], [2:Klein et al.]). In diesem Kapitel soll dennoch eine kompakte Darstellung der charakteristischen Eigenschaften erfolgen, damit bei der späteren Diskussion der Evaluationstechniken und der Ergebnisse Bezugnahmen auf das INVISIP Design nachvollzogen werden können.

Die in INSYDER ursprünglich separat angebotenen einzelnen Visualisierungen wie ResultTable, BarGraph oder SegmentView wurden bei INVISIP in eine gemeinsame Visualisierung namens „LevelTable“ integriert, die die Treffermenge in tabellarischer Form darstellt, dabei aber eine viel weitergehende Funktionalität anbietet (siehe 2.2), als es beispielsweise die Listendarstellung von Suchtreffern bei Web-Suchmaschinen erlaubt.

Als Weiterentwicklung und Alternativ-Konzept zur „LevelTable“ wurde die „GranularityTable“ entworfen, deren Besonderheiten in 2.3 näher erläutert werden. Neben LevelTable und GranularityTable stellt der „Scatterplot“ (siehe 2.4) eine weitere wichtige Möglichkeit zur zweidimensionalen Darstellung der Treffermenge dar.

Alle drei Komponenten sind nach dem Prinzip des Brushing & Linking miteinander verbunden und können durch den Benutzer ergänzend eingesetzt werden. In Abbildung 2-10 ist zur Illustration die LevelTable (obere Hälfte) und der Scatterplot (untere Hälfte) - wie sie in dem zu evaluierenden Mock-Up integriert waren - dargestellt. Auf die Kopplung und Interaktion zwischen beiden Komponenten wird in 2.4 näher eingegangen.

Die im Folgenden dargestellten Grafiken und Designkonzepte entsprechen dem Entwicklungsstand des Metadaten-Browsers im Zeitraum der durchgeführten Evaluationen im Projektpraktikum. Es sind Screenshots der in Kapitel 3 beschriebenen Mock-Ups.

Auch wenn der heutige Projektstand von INVISIP sich im Erscheinungsbild und in der konkreten Implementation teilweise von der folgenden Darstellung deutlich unterscheidet, sind die grundlegenden Designprinzipien erhalten geblieben und neue Designentscheidungen unter Berücksichtigung der hier besprochenen Evaluationsergebnisse getroffen worden.

2.2 Das LevelTable Design

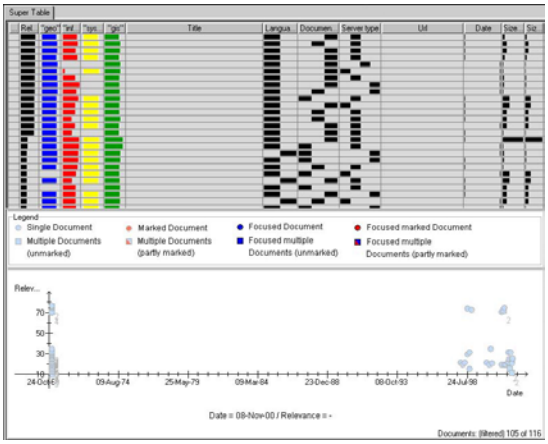

Abbildung 2-2: LevelTable (Level 1)

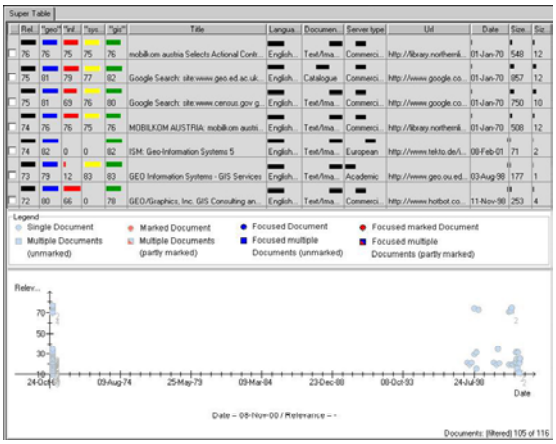

Abbildung 2-1: LevelTable (Level 2)

In Abbildung 1 ist die LevelTable im Grundzustand abgebildet (Level 1). Die einzelnen Zeilen der tabellarischen Darstellung entsprechen jeweils einem Treffer (einem Dokument). Die Spalten enthalten dabei die ermittelten Relevanzen oder die Informationen aus dem jeweiligen Metadatensatz. In diesem Beispiel wurde eine Suchanfrage bezüglich dreier Schlüsselworte abgeschickt. In den Spalten 2 bis 6 sind die Gesamtrelevanz und die Relevanz für die einzelnen Suchworte (zur Unterscheidung wird im Metadaten-Browser für die einzelnen Suchbegriffe eine einheitliche Farbkodierung eingesetzt) als Balken dargestellt. Metadatenattribute wie Größe oder Datum sind in den weiteren Spalten visualisiert.

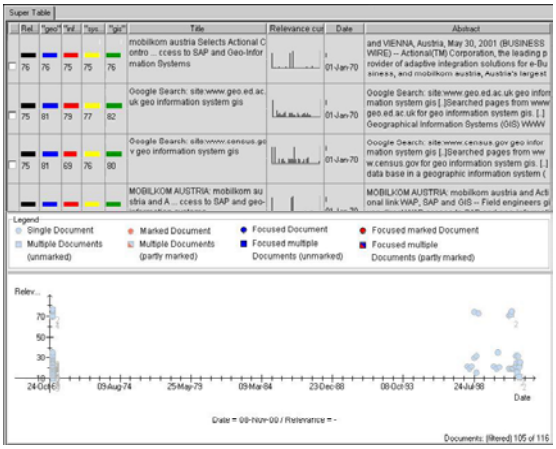

Abbildung 2-3: LevelTable (Level 3)

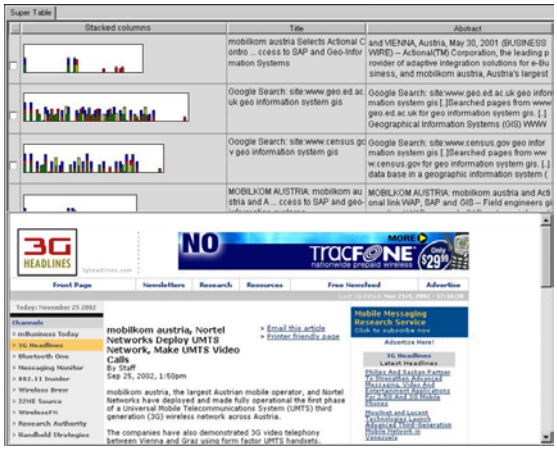

Abbildung 2-4: LevelTable (Level 4)

Neben den Interaktionstechniken, wie sie allgemein von Tabellen bekannt sind (z.B. Sortierung durch Klick auf Tabellenheader), gibt es in der LevelTable die Möglichkeit, die Menge der dargestellten Detailinformation über die Dokumente gezielt zu steuern.

Dazu wird allen Zeilen im Dokument über Pushbuttons („Level 1“, „Level 2“ ...) ein „Level of Detail“ auf einer Skala von 1 – 4 vom Benutzer zugewiesen (siehe Abbildung 2-10: LevelTable + Scatterplot oben).

Während auf Level 1 (Abbildung 2-1) noch ausschließlich Balken visuelle Information über die Dokumente liefern (z.B. Größe in KByte), werden auf Level 2 (Abbildung 2-2) Dokumenttitel oder URLs als Text in der Tabelle wiedergegeben, bis schließlich auf Level 3 zusätzlich Abstracts und weiterführende Sub-Visualisierungen erscheinen (Abbildung 2-3). Auf Level 4 (Abbildung 2-4) wird dann die reine Metadaten-Darstellung verlassen und zusätzlich der Content des Dokuments in einer Browser Preview im unteren Teil des Bildschirms anstelle des Scatterplots eingeblendet.

Mit höherem Level of Detail wird die Gesamtzahl der auf einem Bildschirm sichtbaren Dokumente geringer, da sie zugunsten der weiteren Details verkleinert werden muss. Dies ist nötig, da mehr Information auch mehr Raum der limitierten Bildschirmgröße in Anspruch nimmt.

Der Level of Detail ist dabei aber nicht zwangsläufig ein globaler Parameter für die gesamte Tabelle, sondern lässt sich durch Interaktion mit der Maus auch individuell für ein oder mehrere Dokumente erhöhen. Dies erlaubt dem Benutzer die Fokussierung der Tabellendarstellung auf die Details der relevanten Teilmenge der enthaltenen Dokumente („Focus of Interest“).

2.3 Das GranularityTable Design

Der Einsatz von Visualisierungen im Document Retrieval leidet unter den häufigen Wechseln der Modalitäten innerhalb eines Suchprozesses: Vom Formulieren der Suchanfrage bis zum Erhalten des gewünschten Ergebnisses muss mindestens einmal, aber in der Regel mehrmals, der Kreislauf zwischen Anfrageformulierung, Visualisierung der Treffermenge, Selektion von relevanten Dokumenten und Überprüfung der Relevanz der Dokumente durch Sichtung des Content durchlaufen werden [2:Eibl et al.].

Jeder dieser Durchläufe bedeutet also zwei gravierende Wechsel in der Modalität, nämlich von der textuellen Ebene der Suchanfrage hin zu der visuellen Ebene der Darstellung der Treffermenge und von dort aus wieder zurück zur textuellen Darstellung des Dokumentes. Dies ist

gravierend, da der Mehrwert einer Visualisierung in der Ausnutzung der hochentwickelten visuellen Informationsverarbeitung des Menschen liegt [3:Card et al.]. Häufige Wechsel in den Modalitäten zerstören diesen Nutzen durch die wachsende notwendige Transferleistung, die erbracht werden muss, um die visuellen Erkenntnisse von einer Ebene auf die andere Ebene zu übertragen. Da in der Praxis eine Vielzahl von Zyklen zum erfolgreichen Abschluss eines Suchprozesses notwendig sein wird, muss daher die kognitive Belastung für den Benutzer durch weniger oder weichere Wechsel in der Modalität minimiert werden [2:Eibl et al.].

Mit diesem Ziel vor Augen wurde ein zweites Konzept der LevelTable entworfen und prototypisch umgesetzt, das als „GranularityTable“ bezeichnet wird. Während die LevelTable nur die Wahl eines definierten Detailgrades zwischen 1 und 4 ermöglichte, war es bei der GranularityTable das Ziel, eine stufenlose Informationszu- und –abnahme durch die Verwendung von Slidern anzubieten. Die neue im Informationsgehalt stufenlos regulierbare Alternative besitzt somit

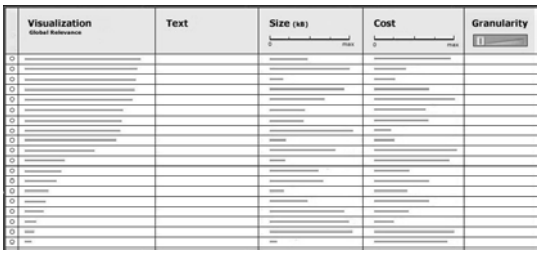

Abbildung 2-5: GranularityTable (kleine Gran.)

Abbildung 2-7: GranularityTable (mittlere Gran.)

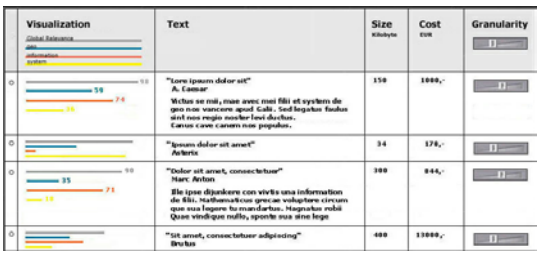

Abbildung 2-6: gemischte Granularity

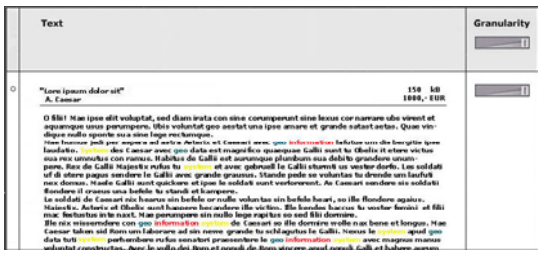

Abbildung 2-8: GranularityTable (max. Gran.)

keine unstetigen Sprünge bei Übergängen im Level of Detail und zwischen Metadaten-Visualisierung und Content-Ebene [2:Klein et al.].

Der Begriff „Granularity“ als Metapher für den Level of Detail wurde dabei aus der Fotografie übernommen, wo die Granularity (=Körnung) des Filmmaterials die Detailgenauigkeit und die Feinheit der Aufnahmen bestimmt.

Bei der GranularityTable lässt sich mithilfe eines globalen Sliders im Tabellenheader die Granularität der gesamten Tabelle wählen und mithilfe lokaler Slider in den individuellen Zeilen für einzelne Dokumente individuell einstellen (Abbildung 2-6 und Abbildung 2-7).

Dabei bedeutet eine hohe Granularität eine hohe „Körnung“ in der Tabellendarstellung, d.h. viel Information (viel Raum) zu einem Einzeldokument - mit geringem oder keinem Überblick über die gesamte Treffermenge - im Gegensatz zu einer Vielzahl angezeigter Dokumente mit nur geringer individueller Information bei kleiner Granularität (Abbildung 2-5).

Dabei wird der konkrete Content der Dokumente mit steigender Granularität sukzessive in die Metadaten-Visualisierung integriert (Abbildung 2-6 und Abbildung 2-7), bis er schließlich am Schluß im Zentrum der Darstellung steht (Abbildung 2-8). Dabei ist der Unterschied zur LevelTable insbesondere, dass diese Art der Browser-Preview nach wie vor Bestandteil der Tabelle und nicht eine isolierte Komponente anstelle des Scatterplot ist und durch das Highlighting von Suchbegriffen oder Navigationshilfen einen Mehrwert gegenüber einer reinen, externen Browser-Darstellung liefert.

2.4 Der Scatterplot

Der Scatterplot (Abbildung 2-9 und Abbildung 2-10 unten) ist eine 2D-Darstellung der Treffermenge, in der jedes einzelne Element als Punkt in die von zwei Achsen aufgespannten Ebene eingezeichnet wird. Dabei werden anstelle eines einfachen Datenpunktes Glyphen verwendet, die weiterführende Informationen über Art und Selektionszustand in Farbe und Form kodieren und mit ihrer Größe und Kontur Angriffsfläche für die Interaktion mit der Maus bieten.

Die Achsen können dabei flexibel mit beliebigen Metadatenattributen oder Relevanzen belegt werden (z.B. Auftragen der Relevanz gegen Größe in KByte - siehe Abbildung 2-9).

Durch sinnvolle Wahl der Achsenbelegung kön-

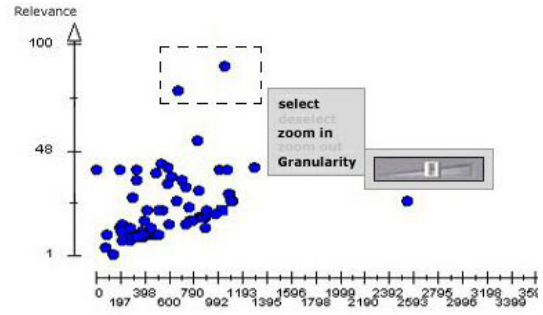

Abbildung 2-9: Interaktion mit Scatterplot

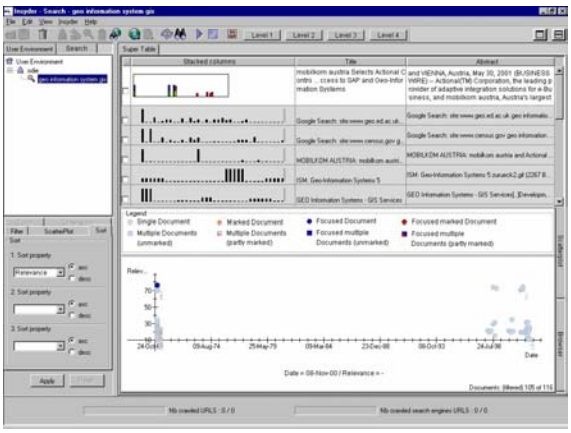

Abbildung 2-10: LevelTable + Scatterplot

nen so beim Betrachten von Formationen und Anhäufungen innerhalb der dargestellten Dokumentglyphen Verwandtschaften zwischen Dokumenten erkannt werden, beispielsweise Cluster mit ähnlichen Relevanzen (d.h. semantische Nähe) oder Linien mit gleicher Sprache oder gleichem Dokumenttyp (d.h. formale Verwandtschaft). Die Scatterplot-Darstellung ermöglicht somit einen alternativen Einblick in die Treffermenge und eine direkte Selektion von Dokumenten oder Dokumenten-Clustern durch Mausklick oder das Öffnen einer Bounding Box.

Mit den Brushing & Linking Techniken des Metadaten-Browsers ist es möglich, mit den Tabellenvisualisierung und dem Scatterplot ergänzend zu arbeiten: Selektionen, die in den Tables vorgenommen werden, werden simultan auch im Scatterplot angezeigt und umgekehrt. Abbildung 2-9 zeigt beispielsweise, wie die Kopplung zwischen Scatterplot und GranularityTable es ermöglicht, die Granularität (siehe 2.3) eines Dokuments zu erhöhen, das im Scatterplot identifiziert und für das dort ein Kontextmenü durch Klick auf seine Glyphen-Repräsentation geöffnet wurde.

3 Evaluation der INVISIP Mock-Ups

Die Hauptmotivation für die Untersuchung der beschriebenen Designkonzepte aus Kapitel 2 lag in der notwendigen Überprüfung neuer Bestandteile vor der Realisierung in einer Java-Applikation. Während die LevelTable und der Scatterplot in einer ersten, frühen Evaluation mit Papierprototypen vor Beginn des hier behandelten Zeitraums bereits grundsätzlich bestätigt wurden, waren neue Komponenten wie die GranularityTable bislang nicht durch Evaluation auf ihre Usability hin untersucht worden. Auch die Integration beider Table-Visualisierungen in einen übergeordneten Rahmen namens „SuperTable“, sowie deren Kopplung mit dem Scatterplot sollten dabei auf dem Prüfstand stehen. Die grundsätzlichen Kernfragen sind in 3.1 dargestellt.

Zu Beantwortung dieser Kernfragen wurde für Oktober 2002 eine Serie von Usability-Tests mit Mitarbeitern der Projektpartner als Testpersonen angesetzt (siehe 3.5). Offen blieb zunächst die Frage, wie die dort erhaltenen Erkenntnisse durch zusätzliche Evaluationen erweitert werden könnten. Nach der Diskussion verschiedener Varianten (siehe 3.6.2), wurde schließlich die Durchführung einer webbasierten Usability-Evaluation, wie sie in 3.6 beschrieben ist, beschlossen.

Hintergrund dieser Entscheidung war es, die hohe Aussagekraft des Usability-Tests im Labor mit den Erkenntnissen aus einer zusätzlichen Webuntersuchung über die Zielgruppe und die Designkonzepte zu ergänzen. Weiterhin konnten so auch die Projektpartner über das Web an der Entwicklung partizipieren, die aufgrund des finanziellen und organisatorischen Aufwands nicht oder nur teilweise an einer Laborevaluation in Konstanz teilnehmen konnten (siehe 3.6.1).

3.1 Kernfragen an die Evaluation der Mock-Ups

Die Kernfragen, die mithilfe der Evaluation beantwortet werden sollten, bezogen sich vor allem auf die beiden eingesetzten Varianten LevelTable und GranularityTable.

Primär stellte sich die Frage, ob die positiven Resultate, die die Evaluation des LevelTable Konzepts mit den Papier-Prototypen lieferte, in einer Evaluation mit real wirkenden Prototypen und authentischer Funktionalität bestätigt werden. Sollte dies der Fall sein, bleibt zu prüfen, ob sich die Weiterentwicklung zur GranularityTable genauso bewährt und ob signifikante Verbesserungen oder Verschlechterung gegenüber der LevelTable zu beobachten sind. Ein direkter Vergleich beider Design Entwürfe sollte zwar nicht dazu dienen, ein einzelnes überlegenes Design zu ermitteln, aber die Stärken und Schwächen beider Ansätze zu identifizieren und gegenüberzustellen. Dabei sollte auch überprüft werden, ob die den unterschiedlichen Table-Designs zugrundeliegenden Use Cases (browsing-orienterte vs analytische Suche) auch in der Testpraxis bestätigt werden.

Bezogen auf das Verständnis von LevelTable und GranularityTable stellte sich die grundsätzliche Frage, ob die Testpersonen in der Lage sind, ein mentales Modell aufzubauen, in dem Konzepte wie Relevanz, Level of Detail oder Focus of Interest enthalten sind und auch im Sinne des Designkonzepts richtig verstanden werden. Konkret zu überprüfen war dabei zunächst, ob die Bedeutung der Tabellendarstellung erkannt wird und von den Einträgen die jeweilige Relevanz des dargestellten Dokuments zur Suchanfrage abgelesen werden kann. Ohne Verständnis dieses elementaren Zusammenhangs ist eine gezielte Arbeit mit der SuperTable nicht denkbar.

Als Indikator dafür, ob Level of Detail bzw. Focus of Interest verstanden worden sind, sollte überprüft werden, wie mit den entsprechenden Funktionen umgegangen wird und ob die Unterscheidung zwischen der Wahl des globalen und lokalen Level of Details (siehe z.B. Abbildung 2-6: gemischte Granularity) auf Userseite gelingt. Auch mögliche Probleme beim Übergang von der reinen Metadaten-Darstellung zur Content-Ebene und bei der low-level Interaktion mit der Tabelle (Selektion und Deselektion von Dokumenten, Platzierung des Fokus, Verändern der

Sortierreihenfolge) waren im Rahmen der Evaluation zu klären, wobei sie gegenüber der Thematisierung der grundsätzlichen Konzepte im Hintergrund standen.

Neben den Tabellendarstellungen sollte auch der Scatterplot in seiner Bedeutung für die Arbeit mit dem Metadaten-Browser überprüft werden. Zu ermittelnde Faktoren waren dabei das Grundverständnis für die Scatterplot-Visualisierung an sich und für die Kopplung zwischen Scatterplot und LevelTable bzw. GranularityTable. Es stellte sich dabei insbesondere die Frage, inwieweit die Testbenutzer Scatterplot und SuperTable als integrierte Komponenten zur ergänzenden Nutzung verstehen würden oder ob beide als voneinander isoliert betrachtet werden.

Als nicht notwendig und als wenig aussagekräftig wurde das Messen von Reaktionszeiten und das ermitteln eines Leistungsindex eingeschätzt. Das Ermitteln eines quantitativen Leistungsindex für ein Designkonzept ist nur dann sinnvoll, wenn ein bereits weitgehend einsatzfähiges System mit vollständigen Interaktionsmöglichkeiten und realistischen Szenarien vorliegt. Ist dies gegeben, kann durch Vergleich mit anderen Systemen eine quantitative Einschätzung der Usability z.B. über die Anzahl der gelösten Testaufgaben vorgenommen werden. Im Falle von INVISIP - bei dem zunächst noch grundlegende Konzepte und keine fertige Applikation untersucht wurden - wäre weder ein valider Leistungsindex ermittelbar gewesen, noch wäre ein Referenzrahmen für Vergleiche mit diesem Index durch andere Untersuchungen vorhanden. Einzige Anwendungsmöglichkeit für eine quantitative Analyse ist der Vergleich von LevelTable und GranularityTable, der dann auch teilweise über Vergleich des Benutzerverständnisses für beide Entwürfe durchgeführt wurde. Insgesamt stellten sich aber im hier behandelten Projektzeitraum eher grundlegende qualitative und keine quantitativen Fragen.

3.2 Anforderungen an die Mock-Ups

Betrachtet man die Fragestellungen in 3.1 wird deutlich, dass es Ziel der Evaluation sein musste das bisherige Designkonzept sowohl auf abstrakter Verständnisebene (Werden die grundlegenden Konzepte verstanden? Entspricht das mentale Modells des Benutzers dem Designkonzept?) und auf Ebene der Interaktion (Gibt es Probleme bei der low-level Interaktion? Ist die Vorgehensweise klar?) zu prüfen. Zusammen mit den Vorgaben zum Zeitpunkt und Umfang der Evaluation ergaben sich daher folgende Anforderungen bei der Entwicklung der Mock-Ups:

1. Es ist notwendig, dass die Mock-Ups die Fähigkeit zum Durchführen der wesentlichen Arbeitsabläufe besitzen, damit das Verständnis der Testpersonen für grundlegende Kon-

zepte und ihr mentales Modell im Rahmen eines durchgespielten Suchprozesses untersucht werden kann.

2. Neben einem wirklichkeitsgetreuen Erscheinungsbild der Oberfläche muss die Funktionsweise der Interaktionselemente weitgehend verbreiteten GUI-Systemen (z.B. Java-Swing Widgets) entsprechen, um einen möglichst hohen Realismus der low-level Interaktion während des Tests zu erreichen.

3. Der Aufwand zur Entwicklung der Mock-Ups muss im sinnvollen Verhältnis zum Aufwand der realen Implementierung stehen. Es muss daher möglichst umfassend mit grafischen Werkzeugen zur schnellen Erstellung der Prototypen gearbeitet werden und wenig bis gar nicht mit Entwicklungsumgebungen zur Entwicklung von realen Applikationen (also „Rapid Prototyping“ anstatt Programmierung).

4. Die Mock-Ups müssen in einem plattformunabhängigen Datenformat vorliegen, das keine Kompilierung oder aufwändige Installation auf dem Testsystem erfordert, insbesondere mit Hinblick auf die Integration in einer webbasierten Evaluation.

3.3 Gestaltung der Mock-Ups

Vor dem Hintergrund dieser Anforderungen und aufgrund der Tatsache, dass bereits zu Präsentationszwecken verschiedene auf HTML- bzw. JPG-Dateien basierende Abbildungen des Design-Konzepts vorhanden waren, wurde schließlich der Entschluss getroffen, zur Entwicklung der Mock-Ups mit HTML verlinkte Grafikelemente zu verwenden. Diese Grafikelemente wurden entsprechend den vorgegebenen Designkonzepten mithilfe von Zeichenprogrammen erstellt und ihnen wurde mit Abbildungen von Java-Widgets ein realistisches Look & Feel verliehen.

Durch das Hinterlegen aktiver Flächen wie den dargestellten Interaktionselementen (Buttons, selektierbare Objekte) mit Hyperlinks, die dann jeweils zu neuen HTML-Dateien mit entsprechend veränderten grafischen Inhalten führten, konnte somit der Eindruck einer interaktiven, kompletten Applikation vermittelt werden. Insbesondere bei der Verwendung der Fullscreen-Darstellung, wie sie bei aktuellen Versionen von gängigen HTML-Browsern wie Microsoft Internet Explorer, Netscape Navigator oder Opera integriert ist, erscheint so der Mock-Up bildschirmfüllend und vermittelt der Testperson einen sehr realistischen Eindruck einer wirklichen Applikation.

Problematisch bleibt dabei jedoch die Flexibilität der Interaktion: Die verschiedenen Systemzustände sind als grafische Kulissen in der Form eines einfachen Verzweigungsbaumes von Hyperlinks seriell hintereinander geschaltet, so dass Raum zur wirklich freien Exploration durch die Testpersonen nur sehr bedingt vorhanden war. Je nach Aufwand bei der Entwicklung des Mock-Ups muss mehr oder weniger stark eine vordefinierte Abfolge von Interaktionsschritten erfolgen. Werden diese vorgegebenen Pfade verlassen, kann es zu „Sackgassen“ innerhalb der Interaktion kommen. Im Falle der INVISIP Mock-Ups war daher die Rückkehr zum Ausgangszustand während einer freien Exploration oft nicht möglich, wie auch später im Labor durch viele notwendige Interventionen der Testleitung deutlich wurde.

Ebenfalls problematisch war die Vielzahl von realistisch abgebildeten Interaktionselementen (Icons zum Fenstermanagement, Slider, Radiobuttons, Checkboxes etc), die nicht mit Funktionalität hinterlegt waren, sondern nur zur Wahrung eines vertrauten Erscheinungsbildes dienen sollten. Diese führten selbst bei erfahrenen Testpersonen zu Irritationen, da die Unterscheidung zwischen wirklich interaktiven Elementen des Prototyps (also den hinterlegten Hyperlinks), der dargestellten „Windows-Kulisse“ im Prototyp und der realen Windows-Oberfläche auf dem Testrechner schwer fiel. Dieser Effekt - wie er auch aus aggressiver Bannerwerbung im Web bekannt ist, bei der Fensterdarstellungen oder Pushbuttons als GIF-Grafiken in Websites integriert werden, um den Benutzer zu einem unbeabsichtigten Klick auf einen Hyperlink zu bewegen - machte eine unbetreute Exploration der Mock-Ups zusätzlich schwierig.

Ein weiterer nicht unerheblicher Nachteil bei der Verwendung von HTML Prototypen ist die eingeschränkte Mausinteraktion. Während sich das Drücken von Pushbuttons oder die Selektion von Objekten, Checkboxes oder Radio-Buttons problemlos darstellen lässt, ist der Doppel- oder Rechtsklick nicht abzubilden. Der Rechts-Klick führt zwangsläufig zur Öffnung eines Kontext-Menüs des Browsers, da ja die gesamte Interaktion mit dem Mock-Up innerhalb eines HTML-Browsers abläuft. Der Doppelklick ist ebenfalls nicht umzusetzen, da nach allgemeiner Konvention die Mausinteraktion im Web nur auf einfachen Mausklicks basiert und daher schon beim ersten Klick aktiv wird. Dementsprechend können auch keine Drag-and-Drop Operationen in die Mock-Ups integriert werden.

Im Falle von INVISIP, wo in einer frühen Projektphase vor allem die grundlegenden Konzepte untersucht wurden, waren dies akzeptable Abstriche. Es gelang trotzdem, mithilfe des HTML-Prototyping schnell die ausreichende Funktionalität in den Mock-Ups anbieten zu können. Für die Fragestellungen aus 3.1 war der erreichte Grad des Realismus ausreichend. Je mehr jedoch die low-level Interaktion und die Details der Implementierung in den Mittelpunkt rücken, desto

stärker fallen diese Einschränkungen ins Gewicht und werden schließlich zu Einschränkungen im gesamten Nutzen eines HTML-Prototypen.

Im Falle von INVISIP standen schon kurz nach der Evaluation der Mock-Ups eine erste Version der Java-Applikationen zur Verfügung (siehe Kapitel 4), weshalb sich hier die in Kapitel 5 vorgestellten Ansätze der Evaluation eignen.

Bei anderen Projekten und unterschiedlicher Zielsetzung der Evaluation könnte es jedoch sinnvoller sein, wieder auf Mock-Ups und die immer populärere Macromedia Flash Technologie zur Schaffung von interaktiven Prototypen zurückzugreifen. Diese teilt sich mit HTML den Vorzug plattformunabhängig und vollständig über das Web zugänglich zu sein, bietet dabei aber weitaus mehr Möglichkeiten bei der schnellen Entwicklung von komplexen Prototypen mit leicht beherrschbaren visuellen Werkzeugen. [6:Ludi].

3.4 Befragung der Zielgruppe

Neben dem Metadaten-Browser und dem Verständnis der Testpersonen für die Oberfläche war bei der Evaluation auch die Test- und Zielgruppe an sich von Interesse.

Es ist für die Bewertung einer Laboruntersuchung unerlässlich mehr über die Testpersonen, ihre Fähigkeiten und ihren Hintergrund zu erfahren, um die Erkenntnisse aus dem Labor auch entsprechend gewichten zu können. Aus diesem Grund werden vor und nach der Testsitzung von der Testperson Pre-Test- und Post-Test-Fragebögen ausgefüllt, die versuchen, sowohl objektive Eigenschaften (Vorkenntnisse, PC-Erfahrung) wie auch subjektive Einschätzungen der Testperson (persönliche Einschätzung gegenüber dem Test/dem Testprodukt, emotionaler Zustand, ...) in Erfahrung zu bringen.

Diese Fragebögen können ein deutlicher Hinweis auf die Validität einer Untersuchung sein: Wenn beispielsweise ausschließlich erfahrene PC-Nutzer eine Software testen, die eigentlich für die Allgemeinheit konzipiert ist, können diese aufgrund ihrer Praxis unter Umständen ein unrealistisch gutes Ergebnis erzielen. Die Untersuchung als solche ist aber für das Projekt wertlos, da ein gutes Ergebnis unter PC-Profis nur wenig darüber aussagt, wie das Ergebnis bei den eigentlich angepeilten Anfängern ausgefallen wäre.

Die Auswahl der richtigen Testpersonen ist daher von elementarer Bedeutung für die Gültigkeit der Evaluationsergebnisse im realen Anwendungskontext eines Produktes. Hier spielt die Kenntnis der Zielgruppe (siehe unten) und das richtige Stichprobendesign, wie es von der empi-

rischen Sozialforschung erschöpfend behandelt wird, eine große Rolle (als Einführung: [6:Schnell et al., pp. 247-251]).

Die bei INVISIP verwendeten Pre- und Post-Test-Fragebögen deckten sich zwecks besserer Vergleichsmöglichkeit weitgehend mit den Teilen der später durchgeführten Webumfrage, die zur Ermittlung des Userprofils dienten und können Anhang B 7.2 entnommen werden.

Im Falle von INVISIP waren alle Teilnehmer der Labor- oder Web-Evaluation in der Entwicklung oder Nutzung von GIS involviert. Diese bewusste Entscheidung für ein Fachpublikum wurde vor dem Hintergrund der Anwendungsdomäne des Metadaten-Browsers getroffen und entsprach dem zugrundgelegten Anwendungsszenario. Dennoch sollte starken Verzerrungen durch das Einholen von Informationen über die berufliche Tätigkeit, über Computerkenntnisse und die Verbindung zum Projekt entgegengewirkt werden.

Um zu überprüfen, ob bei der Laboruntersuchung eine repräsentative Stichprobe der Zielgruppe herangezogen wurde, muss wiederum klar sein, wer die Zielgruppe darstellt und welche Eigenschaften innerhalb der Zielgruppe vertreten sind. Um dieses Bild von der Zielgruppe von INVISIP zu gewinnen, diente vor allem die Befragung im Rahmen der webbasierten Usability-Evaluation (siehe 3.6), die aufgrund ihrer größeren Stichprobengröße (n = 32) ein genaueres Bild der Nutzer aus der Anwendungsdomäne entwerfen konnte. Mit diesen Erkenntnissen ließen sich dann die Erkenntnisse aus der Laborevaluation (n = 8) als mehr oder weniger repräsentativ beurteilen.

Die Gelegenheit einer Benutzerbefragung konnte weiterhin dazu genutzt werden, um die bisherige Arbeitsweise, vorhandene Ausstattung und grundsätzliche Akzeptanz von Informationstechnologie innerhalb der Zielgruppe zu ermitteln. Mit Hinblick auf die unterschiedlichen Use Cases von LevelTable und GranularityTable wurde auch eine Befragung zur angewandten Suchstrategie durchgeführt (siehe 3.6.6).

Eine Übersicht über die Inhalte der Webbefragung findet sich in Anhang B 7.2.

3.5 Laborevaluation der Mock-Ups

3.5.1 Eigenschaften des Usability-Tests im Labor

Wie in 3.1 beschrieben war eine der Kernfragen, zu deren Beantwortung die Evaluationen der INVISIP Mock-Ups durchgeführt wurden, wie das mentale Modell der Testpersonen bei der Arbeit mit dem Metadaten-Browser aussieht und inwieweit es sich mit den Designkonzepten aus Kapitel 2 deckt. Der Usability-Test im Labor stellt dabei eine wirksame Methode dar, die vom Benutzer im Geiste generierte Modellvorstellung von einem System über Beobachtung und Befragung zu ermitteln. Mit diesem Einblick in die Denkweise des Benutzers ist es dann möglich, schwer verständliche oder scheinbar unlogische Aspekte im Designkonzept zu ermitteln, konkrete Redesign-Vorschläge zu erarbeiten und somit Usability-Problemquellen auszuräumen [6:Schulz et al.].

Zu diesem Zweck wird die Testperson mit Mock-Ups oder einem lauffähigen System im Usability-Labor konfrontiert, wobei sie vorgegebene Aufgabenstellungen selbstständig mit dem System bearbeitet. Während der Bearbeitung wird sie vom Testleiter betreut, der den Testverlauf so moderiert, dass das mentale Modell der Testperson im Verlauf deutlich wird und dass eine unproduktive Entwicklung des Testverlaufs vermieden wird. Weitere Informationen zur Methodik und zur Validität des Usability-Tests findet sich bei [6:Nielsen 1] und [6:Nielsen 2].

Im Rahmen einer derartigen Untersuchung wird das Userverhalten durch Auswertung von Aufzeichnungen aller Art (Videokamera, Mikrofone, Protokollierung durch Beisitzer, Screencam) nach häufiger auftretenden Verhaltensmustern durchsucht, die dann auf Ungereimtheiten oder Lücken in den mentalen Modellen der Testpersonen hin überprüft werden können.

Eine sehr wichtige Datenquelle ist dabei das „thinking-aloud“, also die Kommentare, die der Testbenutzer zur Erläuterung seiner Aktionen während des Tests abgibt. Auf der Basis dieser Aussagen lassen sich tiefe Einblicke in die Denkweise der Testperson gewinnen und deren Belastung besser einschätzen. Damit diese wichtige Informationsquelle nutzbar ist, ist es sinnvoll, die Testperson während des Tests anzuweisen, ihre Handlungen zu begründen und zu kommentieren, was oftmals auch den Nebeneffekt eines konzentrierteren und aufmerksameren Vorgehens hat.

Auch Körperhaltung, Abstand vom Bildschirm, Mausbewegungen und Mimik der Testperson sind wertvolle Informationsquellen. Beispielsweise kann an ihnen abgelesen werden, wenn die

visuelle Gestaltung eines Systems die Testperson verwirrt. Deutlich wird dies meist durch langsame, zögerliche Bewegungen des Mauspfeils, Heranrücken an den Bildschirm, Fehlklicks oder einen angestrengten Gesichtsausdruck. Das Auftreten solcher Verhaltensweisen sind immer starke Anzeichen für Usability-Probleme und daher gut verwertbare Hinweise bei der Suche nach Verbesserungsmöglichkeiten.

3.5.2 Durchführung der Laborevaluation bei INVISIP

Zu Beginn der Evaluation wurde ein Video gezeigt, das das zugrunde liegende Konzept hinter beiden Prototypen erläuterte. Nach dieser Einweisung wurden die vorbereiteten Aufgaben nacheinander einzeln den Testpersonen vorgelegt. Während der Durchführung der Aufgaben wurden Rückfragen zur konkreten Bedienung, deren Beantwortung das Ergebnis der Evaluation zu verzerren drohte, nicht beantwortet. Prinzip war es, die Testperson völlig allein mit den Mock-Ups arbeiten zu lassen, um damit ein möglichst objektives Bild der Arbeit zu erhalten.

Um Verzerrungen durch Lerneffekte innerhalb einer Testsitzung auszuschalten wurde die Reihenfolge, in der die verschiedenen Prototypen präsentiert wurden, von Testperson zu Testperson variiert (siehe dazu auch 3.6.5).

Die Testpersonen wurden bei der Evaluation im Usability-Labor mithilfe einer Videokamera mit Mikrofon frontal vor dem Testrechner aufgezeichnet, so dass „thinking-aloud“, Mimik und Körperhaltung des Benutzers komplett erfasst wurden. Weiterhin wurde eine Screencam-Software zur Aufzeichnung des Geschehens auf dem Display des Testrechners eingesetzt, um die Mausbewegungen, Klicks und einzelnen Interaktionsschritte genau zu erfassen.

Neben der technischen Aufzeichnung führte auch ein Beisitzer handschriftlich Protokoll über die Handlungen und Reaktionen der Testperson, was sich für die spätere Auswertung aufgrund der hohen Informationsdichte empfiehlt. Da die komplette Sichtung von Videomaterial sehr viel Zeit erfordert, kann ein gutes Protokoll die Auswertung erheblich beschleunigen, da die Video-Sichtung auf die Ereignisse reduziert wird, die im handschriftlichen Protokoll als relevant aufgeführt sind.

Der Umfang einer Testsitzung variierte zwischen 45 Minuten und einer Stunde. Die Möglichkeit zum Abbruch des Tests oder zu einem verfrühten Ende nahm dabei keine der Testpersonen in Anspruch. Wie in 3.4 erwähnt, wurde zu Beginn und zum Ende jeder Testsitzung ein Pre-Test- bzw. Posttest-Fragebogen vorgelegt, der Zusatzinformationen zur Testperson sammelte und half, die Erkenntnisse aus der Sitzung einzuordnen.

3.5.3 Testmoderation

Bei der Durchführung des Usability-Tests im Labor ist die Fragestellung an die Evaluation entscheidend für die Art und Weise wie die Aufgaben gestellt, der Test moderiert und wie der Testverlauf aufgezeichnet werden sollte. In 3.1 wurde bereits diskutiert, dass die Evaluation im Rahmen von INVISIP nicht zur Ermittlung eines Leistungsindex oder anderer quantitativer Indikatoren für die Usability des Designkonzepts dienen soll, was die Durchführung deutlich vereinfacht.

Während bei einer komparativen Usability-Evaluation mit quantitativen Fragestellungen Verzerrungen und Störfaktoren durch Aufbau, Art der Moderation oder Aufgabenstellung mit präziser Planung minimiert werden müssen, konnte bei der hier behandelten Laborevaluation mehr Flexibilität und eine lockerere Testkonzeption zum Einsatz kommen. Es wurde dabei angestrebt, die jeweiligen Fragenkomplexe, die an die Evaluation gestellt wurden, für jede Person umfassend zu beantworten, indem hypothetische Suchprozesse anhand der vorgegebenen Aufgabenstellungen durchgespielt wurden. Der Test musste dazu aber nicht streng in der vorgegebenen Reihenfolge durchgeführt werden, sondern es bestand die Möglichkeit, Aufgaben zunächst abzubrechen und mit anderen fortzufahren. Auch der Dialog zwischen Testleiter und Testperson war nicht ausschließlich auf das Mitteilen der Aufgabenstellung beschränkt. Bei interessanten Aussagen der Testperson, war es der Leitung erlaubt, nachzufragen, um persönliche Einschätzungen und den Verständnisgrad zu ermitteln. Somit konnten umfassende und zusammenhängende Informationen über das mentale Modell und die beabsichtigte Arbeitsweise der Testpersonen gesammelt werden.

Diese Vorgehensweise war vor dem Hintergrund der vor allem qualitativen und kaum quantitativen Fragestellungen an die Evaluation berechtigt. In anderen Kontexten wäre dies jedoch ein grober Verstoß gegen eine objektive Testmethodik, z.B. beim Vergleich der Effizienz konkurrierender Buchungssysteme, bei dem die Zahl der bearbeiteten Vorgänge als Leistungsindex herangezogen wird. Eine freie Gestaltung des Testablaufs würde hier die ermittelten Resultate unter Umständen massiv beeinflussen.

Dennoch war der Verlauf des Tests durch die vorgegebenen Aufgabenstellungen in einen klaren formalen Rahmen eingebettet, d.h. auf alle Kernfragen aus 3.1 wurde durch Bearbeitung der vorgegebenen User-Tasks in ausreichender Intensität eingegangen. Dies war für eine formale Auswertung der Daten zwingend notwendig. Ein zu dialog-ähnlicher Testverlauf ohne festen inhaltlichen Rahmen würde eher der Evaluation mit „constructive interaction“ ähneln. Deren

Ergebnisse sind stark von der Beurteilung des Dialogs durch den Auswertenden abhängig und wären mit der Testleitung als Dialogpartner ohnehin in inakzeptablem Maße beeinflusst.

Um einen Eindruck von der Vorgehensweise beim Usability-Test zu gewinnen, sind in Anhang A 7.1.1 und 7.1.2 die Aufgabenstellungen und die Bemerkungen für die Testleitung aufgeführt.

3.5.4 Auswertung

Nach der Testdurchführung wurde das gesammelte Videomaterial zum besseren Zugriff digitalisiert und auf CD-ROMs archiviert. Es erwies sich als sehr wirksam, bei der Auswertung beide Videoaufzeichnungen von Screencam und Testperson synchron abzuspielen, da dadurch ein sehr umfassender Eindruck sowohl von der Interaktion, als auch vom Benutzer und seinen Absichten vermittelt wird, was gerade bei kritischen und interessanten Stellen innerhalb der Evaluation von großem Vorteil war.

Weiterhin wurden die Angaben in den Pre-Test- und Post-Test-Fragebögen zur Auswertung in Microsoft Excel übertragen und standen von da an zur weiteren Verarbeitung oder zum Export in Statistiksoftware zur Verfügung.

Die Auswertung der Testaufzeichnungen wurde zunächst für die LevelTable, dann für die GranularityTable durchgeführt. Durch das Heranziehen der schriftlichen Protokolle wurden für jede einzelne Person die Vorgänge während der Testsitzung anhand der Videoaufzeichnungen nachvollzogen und alle relevanten Ereignisse und Äußerungen erfasst.

Eine erste rohe Sammlung von Kommentaren der Testpersonen konnte bereits ganz ohne formale Auswertung als reichhaltige Quelle für Hinweise auf auftretenden Usability-Probleme und als Inspiration für Redesign-Vorschläge und weitere Features genutzt werden. Kommentare oder Fragen wie z.B. „Ist die Granularity denn stetig?“ lassen schon vor einer genaueren Auswertung auf Verständnisprobleme oder problematische Konzepte im Entwurf schließen.

Als Beispiel für eine Inspiration durch die Testperson ist hier der Kommentar eines Benutzers zu nennen, der sich enttäuscht darüber äußerte, dass keine Möglichkeit zum Springen zwischen den speziell markierten Suchbegriffen innerhalb eines Textdokuments besteht, was als neuer Gedanke in das zukünftige Design übernommen wurde.

Auch immer wieder auftretende Verhaltensweisen, wie das Heranrücken an den Bildschirm zu bestimmten Zeitpunkten geben ohne weitere Auswertung bereits Aufschluss darüber, dass auf dem Bildschirm offensichtlich zu kleine Schriftgrößen oder ungünstige Farbkontraste verwendet

werden. Aufgrund der schlechten Lesbarkeit verkleinerten daher die Testpersonen den Abstand zum Display.

Um die Ergebnisse der Evaluation jedoch besser quantifizieren zu können, ist es notwendig, ein formales Gerüst zur Einordnung einer Testsitzung zu schaffen. Ein solches Gerüst für die Auswertung einer Testsitzung mit der GranularityTable ist exemplarisch in Anhang A 7.1.3 aufgeführt.

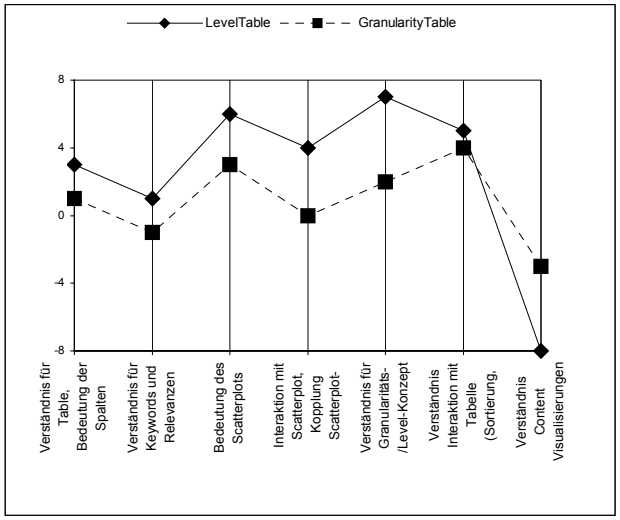

Abbildung 3-1: Verständnisfaktoren (-8 bis 8) für SuperTable

Hier wurden auf der Basis des handschriftlichen Protokolls, der Videoaufzeichnung und der Anzahl der gelösten Aufgaben sieben Faktoren für die Sitzung extrahiert, die auf einer einfachen Skala von „-“, „0“ oder „+“ bewertet worden sind. „+“ bedeutet dabei, dass der Aspekt des Mock-Ups verstanden worden ist, „-“ weist auf komplettes Unverständnis hin.

Wenn also im Beispiel im Anhang A der Faktor „Bedeutung des Scatterplots“ mit „+“ bewertet worden ist, so bedeutet dies, dass die Testperson die Bedeutung des Scatterplots während des Testverlaufs verstanden hat. Neben den anderen Faktoren, deren Bedeutung auch Anhang A entnommen werden kann, war ein weiterer Faktor „Interaktion mit Scatterplot“, in dem das Verständnis für die Beziehung zwischen SuperTable und Scatterplot zusammengefasst wurde. Hier konnte im Beispiel nur ein „-“ erreicht werden.

Aus beiden Faktoren lässt sich somit für die Testperson schliessen, dass zwar der Scatterplot an sich verstanden, aber seine Integration in die Gesamtoberfläche nicht nachvollzogen wurde.

Auf der Basis solcher Auswertungen ist auch eine zusammenfassende Gegenüberstellung der LevelTable und der GranularityTable für alle Testpersonen möglich. Abbildung 3-1 zeigt ein Diagramm in dem jeder der sieben Faktoren für beide Designs und für alle Testpersonen (zusammengefasst durch Addition der Einzelwertungen) dargestellt ist. Dabei sollte die quantitative Komponente der Darstellung nicht überschätzt werden. Als Darstellung des grundsätzlichen Trends und zum Vergleich einzelner Eigenschaften ist das Diagramm jedoch verwendbar. Die häufige Überlegenheit der LevelTable wurde dabei auch in der späteren webbasierten Evaluation bestätigt. Nur im Faktor „Verständnis für Content Visualisierungen“ war die LevelTable der

GranularityTable unterlegen (ganz rechts). Eine ausführlichere Diskussion der Ergebnisse findet in 3.7 statt.

3.6 Webbasierte Evaluation der INVISIP Mock-Ups

Wie in der Einleitung zu diesem Kapitel erwähnt, kamen im Rahmen von INVISIP auch Methoden der webbasierten Usability-Evaluation zum Einsatz, um den Projektpartnern eine Möglichkeit zur Teilnahme an der Evaluation zu ermöglichen, die nicht vor Ort im Usability-Labor anwesend sein konnten. Weiterhin konnte so auch eine größere Zahl von Personen aus der Anwendungsdomäne als Testgruppe herangezogen werden, als normalerweise im Labor wegen des organisatorischen und finanziellen Aufwands möglich ist.

Ein weiterer Nutzen der webbasierten Usability-Evaluation war die Möglichkeit, eine im Unfang größere und repräsentativere Befragung unter Benutzern der Anwendungsdomäne von INVISIP durchzuführen, um besseren Einblick in bisherige Arbeitsweisen, technische Ausstattung, PC-Kenntnisse etc. innerhalb der Zielgruppe von INVISIP zu erhalten.

3.6.1 Eigenschaften der webbasierten Usability-Evaluation (WUE)

Webbasierte Usability-Evaluation (kurz WUE, auch „Remote Usability-Evaluation“) umfasst alle Verfahren des Usability-Testings, die ohne physische Präsenz der Testperson oder einer menschlichen Testleitung in einem Labor über das Internet als Kommunikationsmedium durchgeführt werden können.

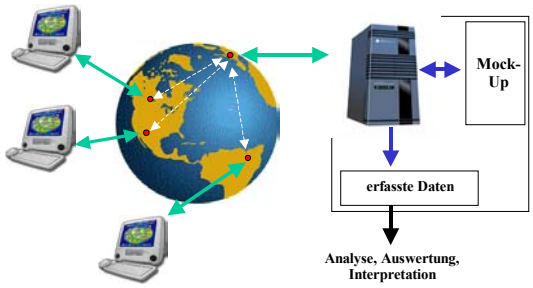

Abbildung 3-2: Schematische Darstellung der WUE

Zu diesem Zweck wird die Präsenz der Testperson mithilfe des Internets im Labor bzw. in einem dafür eingerichteten Server von einem beliebigen Ort bzw. Arbeitsplatz aus abgebildet (siehe Abbildung 3-2). Der Labor-Server (rechts) bietet dabei den zu evaluierenden Prototypen via HTTP, X11 oder Java-Applets zum Zugriff über das Internet an. Dies macht das Untersuchungsobjekt über eine beliebige Workstation (links) zugänglich.

Die Protokollierung und Datenerfassung erfolgt dabei maschinell auf Serverseite. Somit ist die gesamte Testdurchführung automatisiert und damit komplett unabhängig vom Aufenthaltsort der

Testperson oder dem Zeitrahmen der Testleitung. Der große Vorteil der WUE ist daher die annähernd kostenneutrale Auswahl von Testgruppen und –größen: da die Testleitung und Datenerfassung maschinell erfolgt, ist der Umfang, in dem eine WUE durchgeführt wird, unkritisch und somit ist eine starke Erweiterung der Stichprobengröße kaum mit Mehrkosten verbunden. Wegen der Orts- und Zeitunabhängigkeit der WUE müssen keine Terminabsprachen getroffen werden. Reisekosten oder Ausfallszeiten müssen nicht finanziert werden.

Ein weiterer Vorteil, den die WUE in diesem Zusammenhang für INVISIP hatte, war die Möglichkeit, neben den direkt involvierten Projektpartnern auch andere Personen aus der Zielgruppe anzusprechen. Dieses Fachpublikum konnte über die Weitergabe der Webadresse durch die Projektpartner an Kunden oder Geschäftspartner per E-Mail oder durch Ankündigungen auf Fachtagungen gezielt angesprochen werden, wobei dabei die Vorteile des E-Mail-Marketings (keine Versandkosten, niedrigschwelliges Feedback durch simplen Klick auf Hyperlink, Schneeballeffekt innerhalb der Zielgruppe) von großem Vorteil gegenüber dem Versenden von umfangreichen Fragebögen auf dem Postweg mit erfahrungsgemäß geringer Response-Rate waren.

3.6.2 Varianten der WUE für INVISIP

Angesichts der oben dargestellten Vorzüge der WUE wurden im Rahmen des INVISIP Projekts verschiedene Varianten der webbasierten Evaluation diskutiert. Früh kristallisierten sich zwei Varianten als vernünftiger Kompromiss zwischen Entwicklungsaufwand, vorliegenden Prototypen und den zu untersuchenden Usability-Fragestellungen heraus:

1. Das Weblogging mit WebQuilt
2. Die Usability-Webbefragung

3.6.3 Variante 1: Weblogging mit WebQuilt

WebQuilt ist ein Java-Softwarepaket für Windows-Rechner bestehend aus einem Java-Proxy-Server und einem Visualisierungstool für Logfiles, das speziell zur Usability-Evalution von Websites von der Group for User Interface Research an der University of California, Berkeley entwickelt wurde [6:Hong et al.].

WebQuilt fungiert dabei als HTTP-Proxy und kann somit jegliche Interaktion zwischen dem Client-Browser der Testperson und dem Prototyp auf dem Webserver protokollieren, wobei dabei das Augenmerk speziell auf usability-relevanten Daten liegt (siehe Abbildung 3-3). Das Proxy-Konzept erlaubt sogar die Evaluation von Websites, zu denen kein Zugang als Admini-

strator besteht. Es lassen sich beliebige fremde Websites im WWW untersuchen, um beispielsweise eine State-of-the-Art Analyse bei Mitbewerbern vorzunehmen.

Falls gewünscht, kann die Protokollierung für den User dabei völlig unsichtbar stattfinden, so dass er keinerlei Hinweis auf die Aufzeichnung seines Besuches erhält. Umgekehrt können jedoch auch Popup-Fenster mit Informationen zu anstehenden Aufgaben für den User eingeblendet werden, um einen einheitlichen Testverlauf zu realisieren.

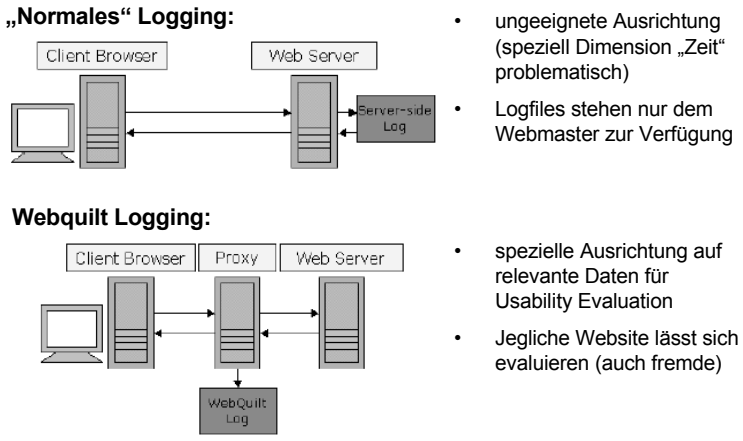

Abbildung 3-3: Normale Logfiles vs. Webquilt Logfiles

Neben dem Logging an sich bietet WebQuilt auch eine Reihe von Werkzeugen zur Visualisierung der erfassten Daten, die eine Auswertung stark vereinfachen. Auf die Möglichkeiten zur Auswertung von Logging Daten wird in [6:Hong et al.] eingegangen.

Aufgrund der technischen Umsetzung und der damit verbundenen begrenzten Information über den Verlauf der Benutzung eines Systems, kann das Weblogging trotz seiner höheren Realitätsnähe nicht mit dem Informationsgehalt einer Laborevaluation mithalten, ist aber dennoch gut geeignet, um gezielt wichtige Eigenschaften eines Systems zu ermitteln. So reicht das von WebQuilt gelieferte Logging aus, um die Navigability einer Site oder einer Menüstruktur zu prüfen. Verweildauern und Navigationsschritte sind mithilfe von WebQuilt messbar und als Graph oder Sitemap visualisierbar. Ein Vergleich mit quantitativen Usability-Goals für das System ist möglich.

Die Einsatzmöglichkeit von WebQuilt besteht selbstverständlich auch für HTML Dateien, die keine Website enthalten, sondern Mock-Ups abbilden (siehe 3.3). Daher mag WebQuilt zunächst als ideales Werkzeug für die Evaluation der Mock-Ups im Rahmen von INVISIP erscheinen. Erste Versuche mit den vorhandenen Prototypen zeigten aber die deutlichen Einschränkungen in der Qualität der Interaktion, wie sie bereits in 3.3 beschrieben worden sind. Diese Einschränkungen führten letztendlich zum Entschluss, dass WebQuilt nur wenig Nutzen bei der Beantwortung der Kernfragen hat, da ohne Moderation und gezielte Befragung der Testpersonen kaum valide Ergebnisse zu erwarten waren.

WebQuilt ist für die Ermittlung von Navigationspfaden und Verweildauern ausgelegt, was für Themen der Web-Usability sinnvoll erscheint. Die dringenden Fragen der Mock-Up Evaluation bei INVISIP lagen aber nicht im Bereich der verwendeten Pfade innerhalb der Mock-Ups, da diese weitgehend vorgegeben waren, und nicht in dem Sammeln von quantitativen Daten (durchschnittliche Verweildauer o.ä.), weil zu diesem Zeitpunkts die Mock-Ups dafür keine ausreichenden Interaktionsmöglichkeiten angeboten haben. Es wurde deshalb der Entschluss getroffen, bei der WUE eine Kombination von Usability-Testing und Userbefragung einzusetzen.

3.6.4 Variante 2: Usability-Webbefragung

Die einfachste Form einer WUE durch Befragung lässt sich mithilfe der Veröffentlichung von allgemein entworfenen Usability-Fragebögen über das Web erreichen. In [5:Perlman] bietet die ACM verschiedene Entwürfe und eine technische Plattform zur Realisierung derartiger Umfragen an. Auch kommerzielle Produkte wie WAMMI werden heute für die Optimierung der Web-Usabiltity im E-Commerce angeboten [5:WAMMI].

Ein großer Nachteil derartiger Userbefragung wird schon in ihrer Bezeichnung als „subjective evaluation“ deutlich. In der Regel erfolgt dabei die Befragung im Anschluss an die Nutzung einer Site, um Aussagen über die Gebrauchstauglichkeit von den Benutzern zu sammeln. Gemessen werden dabei nur subjektiv-empfundene Sympathien und Antipathien gegenüber einem System. Objektiv belegbare Ursachen, der wirkliche vorhandene Grad des Verständnisses für ein System und das zugrundeliegende mentale Modell, das die Testperson entwickelt hat, sind so aber kaum zu erfassen. Diese Kenntnisse sind aber für ein gezieltes Redesign notwendig und standen im Falle von INVISIP zu diesem Zeitpunkt im Vordergrund der Untersuchung.

Bei der „subjective evaluation“ kann es weiterhin zu erheblichen Widersprüchen zwischen tatsächlicher Produktivität und dem subjektiven Empfinden der Benutzer kommen - insbesondere bei der Evaluation von Systemen, die aufgrund ihres Erscheinungsbildes oder äußerer Umstände den Benutzern zunächst besonders attraktiv oder unattraktiv erscheinen. Ein ansprechendes visuelles Design kann gerade bei unerfahrenen Besuchern ein subjektives Empfinden für besondere Leistungsfähigkeit oder hohe Qualität hervorrufen, während nüchtern und minimal gehaltene Oberfläche mit hoher Effizienz als sachlich unterkühlt und daher unattraktiv in der Bedienung empfunden werden [6:Hassenzahl et al.], [6:Davis].

Es war also notwendig durch die Befragung mehr als nur die Sympathie/Antipathie der Testpersonen zu ermitteln, sondern auch objektive Erkenntnisse über deren Verständnis und deren mentale Modelle zu gewinnen, was mit den o.g. vorgefertigten Fragebögen nicht möglich gewesen wäre.

Die Entscheidung fiel zugunsten einer zweiten Variante - der individuell für INVISIP gestalteten Webbefragung. Dabei war es nicht - wie bei konventionellen Usability-Surveys - das Ziel, die Sympathie der Testpersonen gegenüber einem System zu ermitteln, sondern das grundsätzliche Designkonzept in seiner Anwendung durch die Testperson zu überprüfen.

Als Mittelweg zwischen interaktiven Prototypen und einer zu abstrakten Befragung wurde dabei intensiv mit Screenshots und einführenden Videos von den Mock-Ups in Aktion gearbeitet. Ein Screencam-Video von der Benutzung des Systems wurde beispielsweise als einleitende Information in die Befragung integriert, um die grundsätzlichen Konzepte anschaulich zu vermitteln.
Der Fragebogen im Anschluss umfasste hypothetische Suchaufträge, die die Teilnehmer anhand der Screenshots hypothetisch durchspielten und dabei Fragen zur beabsichtigten Benutzung beantworteten. Ein Ausschnitt aus der Befragung ist in Abbildung 3-4 dargestellt.

Naturgemäß waren die Möglichkeiten der hypothetischen Interaktion durch die Antwortvorgaben deutlich eingeschränkter als mit den Mock-Ups im Labor. Dennoch eignete sich diese Technik der Befragung, um das Verständnis der Testperson für einzelne Aspekte der Prototypen gezielt zu überprüfen. Prinzipiell wurde dabei durch die Darstellung der Benutzeroberfläche und durch die Formulierung einer hypothetischen Aufgabenstellung oder Verständnisfrage ermittelt, welche Interaktionsmöglichkeit in der Realität genutzt worden wären. Dies konnte dann bei der Auswertung mit dem Designkonzept verglichen werden.

Das gesamte Verständnis des Benutzers für das dargestellte Produkt – sprich für das mentale Modell – wurde somit über Fragen nach und nach ermittelt und mit dem Entwurf verglichen. Am Ende der Auswertung stand damit für die individuelle Testperson fest, inwieweit ihre Vorstellung von der Umsetzung entfernt ist, die die Entwickler beabsichtigen. Die Distanz ist über die Anzahl der abweichenden Arbeitsschritte und falschverstandener Konzepte ermittelbar und damit auch zur quantitativen Untersuchung geeignet.

Grundsätzliche Problematik bei einer derartigen Untersuchung stellt die Realitätstreue der Befragung dar. Screenshots oder Videos sind nur bedingt geeignet, um einen realistischen Eindruck eines interaktiven Systems bei der Testperson zu erzeugen. Die Testperson muss sich bewusst in

eine hypothetische Situation mit einem nur sehr statisch dargestellten Produkt hineinversetzen, was eine mentale Herausforderung und kognitive Belastung darstellt und somit das Ergebnis verzerren kann. Auch der wichtige Aspekte des Lernens durch Trial-and-Error kann hierbei nicht berücksichtigt werden, da keine Möglichkeit zur Exploration der Oberfläche gegeben ist.

Bewegungsabläufe und Kommentare der Testpersonen, sowie Reaktions- und Verweilzeiten sind nur sehr bedingt oder gar nicht über das Netz übermittelbar. Auf niedriger Ebene auftretende Einzelprobleme - wie z.B. eine irritierende Gestaltung von Icons oder die schlechte Anordnung von Buttons - können somit nicht diagnostiziert werden.

Generell sind solche Userbefragungen zwar in der Lage, grundsätzliche Konflikte zwischen dem zu evaluierenden System und dem mentalen Modells des Benutzers auszumachen, aber bei der Evaluation auf der low-level Ebene (z.B. visuelle Gestaltung, Mausinteraktion) sind sie nicht hilfreich. Während Fehlleistungen auf der Verständnisebene („mistakes“) identifizierbar sind, sind versehentliche Bedienungsfehler auf niedriger Interaktionsebene („slips“) nicht zu diagnostizieren und somit auch nicht abzuschätzen.

Problematisch ist auch die Gestaltung der Fragebögen, die die Beteiligung und die Qualität der Antworten beeinflusst. In [5:Dillman et al.] wird darauf eingegangen, inwieweit die Gestaltung einen Einfluss auf die Beteiligung bei einer Webbefragung hat. Auch Formulierung und Antwortvorgaben sind kritische Faktoren (siehe unten).

3.6.5 Durchführung der webbasierten Evaluation

Es gibt eine Vielzahl von kommerziellen und Freeware Produkten, die Umfragen oder Befragungen auf Webseiten ermöglichen. Gerade im Punkt des Gestaltungsspielraums lassen dabei viele Produkte zu wünschen übrig, indem sie nur sehr starre Formatvorlagen und Frageformate unterstützen oder die Form der Datenspeicherung auf dem Server an bestimmte Datenbanken oder Analysesoftware gekoppelt ist. Da für das INVISIP Projekt jedoch sehr individuelle Typen von Fragestellungen und die Integration von Videos und Screenshots notwendig waren, wurde beschlossen, die Befragung durch vom Autor entwickelte PHP-Skripte und HTML-Templates auf dem Evaluations-Server zu realisieren. Diese Skripte sind seit dem auch für andere Umfragen im Rahmen von INVISIP genutzt worden und können im Anhang B 7.2 in Aktion betrachtet werden.

Neben grundsätzlichen Funktionen zur Darstellung der Fragebögen durch dynamisch generierte HTML-Seiten, zur Speicherung der eingegebenen Daten und zur Verwaltung von konkurrieren-

den Zugriffen durch Session-Management, wurde dabei auch eine Randomisierung der Frageabfolge integriert. Dies war notwendig, da die Präsentation beider Table-Designs in stets gleicher Reihenfolge mit Sicherheit einen positiv-verzerrenden Effekt auf den Grad des Verständnis bei dem zuletzt präsentierten Design gehabt hätte. Wegen der Vorkenntnisse vom Betrachten des ersten Designs, wäre das zweite Design von jedem Teilnehmer viel schneller durchdrungen worden und hätte damit einen unrealistisch positiven Eindruck hinterlassen.

Hauptaugenmerk bei der optischen Gestaltung wurde auf ein schlichtes und möglichst wenig forderndes Erscheinungsbild gelegt, das den Teilnehmer stets über die Anzahl der noch auszufüllenden Seiten informiert, um Frustration und vorzeitigen Abbruch zu vermeiden. Im Falle von INVISIP war es jedoch unvermeidbar während der Befragung häufig zwischen der Darstellung von Screenshots und dem Fragebogen zu wechseln, was zwangsläufig zu einer starken Belastung der Befragten führte. Dementsprechend war auch die ermittelte Fehlerquote bei der hypothetischen Arbeit mit den Mock-Ups im Netz größer als im Labor (siehe 3.7.2).

Die Webumfrage wurde zeitlich begrenzt in einem Zeitraum von 6 Wochen angekündigt und durchgeführt, um zu lange Laufzeiten wegen des Projektfortschritts zu vermeiden, aber auch um Interessierte stärker zur Teilnahme zu motivieren. Als Server diente ein Webserver an der Universität Konstanz auf dem die entsprechende Fragen, die selbstentwickelten PHP-Skripte, die Ergebnisse und die Videos und Screenshots platziert wurden. Die URL des Servers wurde zusammen mit einer Einladung zur Teilnahme an alle Projektpartner mit der Bitte um Weiterleitung an andere Mitglieder der Zielgruppe geschickt. Nach anfänglicher sehr schleppender Beteiligung konnte nach 6 Wochen eine mit 35 Personen ausreichend große Datenmenge gesammelt werden. Anschließend wurden die gesammelten Daten von einigen ungültigen und nur halbausgefüllten Datensätzen befreit und statistisch ausgewertet. Es wurden 32 Datensätze zur Analyse herangezogen.

Anonymität oder physische Abwesenheit kann die Scheu vor negativen Aussagen aufseiten der Testperson verringern. Dies kann entscheidend zur Objektivität der Ergebnisse beitragen, insbesondere wenn wie in diesem Fall, eine engere Beziehung zwischen den Entwicklern, Testleitern und Teilnehmern der WUE besteht. Aber auch negative Effekt durch die anonyme Unverbindlichkeit der Teilnahme sind möglich, beispielsweise wenig Feedback wegen mangelndem Gefühl der Verpflichtung oder schlicht nachlässiges oder bewusst irreführendes Ausfüllen.

Im Falle von INVISIP war dies trotz Anonymität nicht zu beobachten. Bis auf drei klar erkennbare Datensätze wurden keine unvollständigen oder deutlich falsch ausgefüllten Angaben fest-

gestellt. Auch die anfänglich geringe Beteiligung an der WUE konnte ohne eine namentliche Verpflichtung von Projektpartnern und Mitarbeitern auf ein ausreichendes Maß gesteigert werden, indem wiederholt per Mail zur Teilnahme aufgerufen wurde. Effekte wie „lurking“, also dem völlig passiven Besuch der Befragung oder frühzeitiges Abbrechen eines Tests konnten das Testergebnis nicht beeinflussen, da die entsprechenden Erfassungsskripte diese Art der Beteiligung ausblendeten [5:Bosjnak, Tuten].

Die Entscheidung, die WUE anonym durchzuführen, wurde daher bestätigt. Dennoch wurde zum Schutz vor unbefugten Zugriff die Umfrage in einem paßwort-geschützten Bereich des Web-Servers durchgeführt, um die Teilnahme von Personen außerhalb der Zielgruppe zu verhindern.

3.6.6 Beispiel: Ermitteln des Verständnisses für die GranularityTable

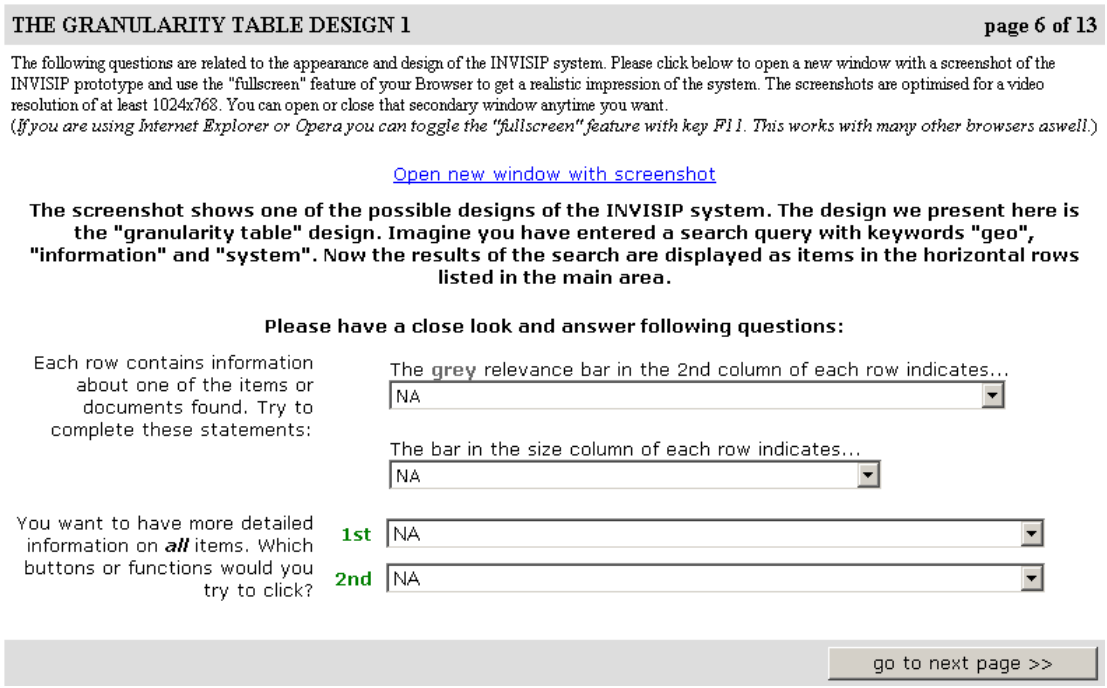

Abbildung 3-4: Screenshot aus der Webbefragung

Im Folgenden sollen anhand einer Beispielfrage aus der Befragung (Abbildung 3-4) die Phasen der Konzeption, Umsetzung und Auswertung der webbasierten Evaluation exemplarisch erläutert werden.

Ausgangspunkt für den oben sichtbaren Entwurf war die Kernfrage nach dem Userverständnis für das GranularityTable Design, nach dem darin verwendeten Konzept der Relevanz und nach der Bedeutung der Balkenvisualisierungen innerhalb der Tabellendarstellung. Weiterhin sollte auf die Wahlmöglichkeiten für die Granularität der gesamten Tabelle bzw. auf die Unterscheidung zwischen lokaler und globaler Granularität (siehe Abbildung 2-6) eingegangen werden.

Da im Gegensatz zum Test im Labor bei der Webbefragung keine direkte Interaktion der Testperson beobachtet werden kann und keine Nachfragen möglich sind, muss die Fragestellung so konzipiert werden, dass deren Beantwortung ein eindeutiges und unmissverständliches Bild vom Userverständnis liefert. Da die Userantworten, die einzige eingehende Information sind, müssen sie klar auswertbar sein und die Fragen möglichst verzerrungsfrei gestellt werden.

„Offene Fragen“ mit Freitexten als Antwort sollten daher nur selten verwendet werden. Besser zu handhaben sind „geschlossene Fragen“ (multiple-choice), deren Antwortvorgaben jedoch ausgewogen gewählt sein müssen [6:Schnell et al., pp. 308 – 312]. Weiterhin müssen die Fragen klar und unmissverständlich formuliert werden (kein „wording bias“), damit keine Beeinflussung durch Formulierung, Wortwahl oder unverständliche Fachterminologie stattfindet [6:Schnell et al., pp. 312 - 317].

Idealerweise wird das Userverständnis so wie in einem Multiple-Choice-Test durch Fragen und Antwortmöglichkeiten abgeprüft, was auch eine Quantifizierung des Verständnisgrades vereinfacht. Dazu werden Antwortvorgaben gemacht, die alle realistisch erscheinen, aber von denen sich nur ein Bruchteil mit dem zu prüfenden Design deckt und als „richtige“ Antwort gilt. Die Antworten, die dem Designkonzept widersprechen, werden als „falsch“ betrachtet. Eine hohe Fehlerquote (also viele „falsche“ Antworten) können dann im Sinne eines Usability-Tests als Mangel im Design - nicht etwa als Unfähigkeit des Benutzers - interpretiert werden.

Hier die konkrete Fragestellung und Antwortvorgaben für die Size-Frage (nur die dritte Antwortvorgabe wird dabei als „richtig“ interpretiert, die anderen als „falsch“). Dabei sollte erwähnt werden, dass zuvor eine Videodemonstration des Designs stattgefunden hat und ein Screenshot der Oberfläche hinzugezogen werden kann.

The bar in the size column of each row indicates...

- ... the size of all documents found
- ... the size of the result of the query
- ... the size of the document displayed in this row
- ... the number of search results

Mithilfe dieses Prinzips lässt sich das Userverständnis für einfache Aspekte eines Designs direkt abfragen. Für jeden Aspekt eines Design kann seine mittlere Fehlerquote für alle Benutzer berechnet werden, die dann deutlich auf die Verständlichkeit dieses Aspektes hinweist.

Abstraktere Konzepte eines Designs können sich durch die Kombination derartiger Fragestellungen untersuchen lassen. Dabei können mehrere einzelne Aspekte abgefragt und mit unterschiedlicher Gewichtung in eine Gesamtwertung einbezogen werden. Beispielsweise impliziert die richtige Beantwortung obiger Frage nicht nur das Verständnis für die Size-Darstellung an sich, sondern auch dafür, dass in einer Zeile jeweils ein Dokument des Suchresultats dargestellt wird. Für eine Bewertung des Gesamtverständnisses für die Tabellendarstellung könnte eine richtige Antwort also mit einer geringen Gewichtung miteinbezogen werden.

Der korrekte Entwurf einer derartigen webbasierten Evaluation zum Abprüfen des mentalen Modells und des Verständnisgrades der Testperson für das zu evaluierende Design stellt sich in der Praxis als schwierig dar. Im Falle von INVISIP wurde bei manchen Fragen beispielsweise die Gewichtung zwischen richtigen und falschen Antworten so ungünstig gewählt, dass eine imaginäre Testperson per Zufallsgenerator eine Trefferquote von 40% erreicht hätte. Dies führte bei der nachträglichen Diskussion der Resultate zu einer sehr geringen Signifikanz der Beobachtungen. Betrachtet man die Ergebnisse in 3.7, so sind einige in ihrer Deutlichkeit teilweise nicht überzeugend und können daher nur als Trendanalyse und grobe Einschätzung herangezogen werden. Soll eine zukünftige Befragung präzisere und besser quantifizierbare Ergebnisse liefern, so wäre ein stärkere Prüfung der statistischen Gesichtspunkte notwendig. Nielsen zeigt in [6:Nielsen 3] jedoch anhand eines vergleichbaren Beispiels zweier konkurrierender Design-Entwürfe, wie Erhebungen bereits bei geringer Signifikanz in der Lage sind, wertvolle Information für das Usability-Engineering zu liefern. Eine Präzision bei der Befragung wie in der empirischen Sozialforschung ist danach nicht zwingend notwendig.

3.7 Zusammenfassung der Ergebnisse der Mock-Up Evaluationen

Als Beleg für den Nutzen der bisher beschriebenen Evaluationen sollen hier in kompakter Form eine Auswahl der Evaluationsergebnisse dargestellt werden. Eine umfassende Darstellung würde den Rahmen dieser Arbeit sprengen, es soll jedoch exemplarisch illustriert werden, wie die gesammelten Informationen über die Zielgruppe von INVISIP, über die Testpersonen und deren mentale Modelle in praxis-relevante Erkenntnisse und Redesign-Vorschläge verwandelt werden können.

3.7.1 Benutzerprofil

Auf der Basis der Webumfrage ergab sich folgendes Bild vom durchschnittlichen Benutzer von GIS und damit der Hauptzielgruppe des INVISIP Metadaten-Browsers: die Gruppe bestand zu 45% aus Männern, zu 38% aus Frauen und 17% machten keine Angaben zu ihrem Geschlecht. 58% der Befragten waren zwischen 30 und 40 Jahren alt. Die Erfahrung im Umgang mit PCs ist unter Benutzern der Anwendungsdomäne sehr groß. 90% haben 5 Jahre oder länger PC Erfahrung, 40% sogar zwischen 11 und 20 Jahren. Die durchschnittlichen Computererfahrung beträgt ca. 10,4 Jahre.

Anhand der Berufs-/Tätigkeitsbeschreibung der Benutzer wurde festgestellt, dass einige Befragte nicht in erster Linie der Anwendungsdomäne von INVISIP entstammen, sondern hauptsächlich in der Software Entwicklung und IT Beratung tätig sind. Daher wurden die Werte für PC Erfahrung um diese Benutzer bereinigt. Trotz dieser Fokussierung auf eine klare GIS Zielgruppe ergibt sich jedoch immer noch eine hohe PC-Erfahrung von durchschnittlich 9 Jahren. Die Erfahrung im Umgang mit dem Internet von allen Benutzern liegt bei einer durchschnittlichen Dauer von 4,7 Jahren. Hier liefert eine Bereinigung eine vernachlässigbar geringe Verringerung auf 4,5 Jahre.

Der prozentuale Anteil der Arbeitszeit, die mit dem PC geleistet wird, liegt bei der Untermenge der INVISIP Zielgruppe im Schnitt bei 68%. Wenn rund 2/3 des Arbeitsalltags vor dem PC bestritten wird, ist zweifellos mindestens eine "basic PC literacy" innerhalb der Zielgruppe vorhanden.

Die Testpersonen wurden gebeten, eine persönliche Bewertung ihrer Erfahrung mit der Arbeit mit dem PC und dem Internet abzugeben, um die Offenheit und die vorhandene Einstellung gegenüber IT Systemen in der Anwendungsdomäne zu ermitteln. Die bereinigte INVISIP Zielgruppe hatte dabei durchweg eine positive Meinung zur Arbeit mit PC und Internet. 90% der Benutzer gaben gute und sehr gute Erfahrungen mit dem PC, 80% gute und sehr gute Erfahrungen mit dem Internet an. Die restlichen wählten einen neutralen Standpunkt. Schlechte oder sehr schlechte Erfahrungen wurden in keinem Fall angegeben.

3.7.2 Vergleich LevelTable vs GranularityTable

Sowohl in der Webumfrage als auch in der Labor Evaluation ergab sich ein einheitliches Bild bezüglich der Erfolgsquote und der Akzeptanz beider Designs:

Beide Untersuchungen ergaben, dass das Verständnis für die Interaktion mit der LevelTable dem GranularityTable Entwurf überlegen ist. Objektiv messbar waren im Labor ein im Schnitt besseres Verständnis für das Level Konzept, mit seinen wählbaren festen Abstufungen im Informationsgehalt, als für die über Slider stetig wählbare Granularität (siehe Abbildung 3-1). Auch das Arbeitstempo lag im Schnitt bei der LevelTable über dem der GranularityTable.

In der Webumfrage wurde bei beiden Tabellen eine generell größere Fehlerquote bei der Interpretation der Bedeutung von einzelnen Spalten und der Nutzung von Interaktionstechniken als im Labor festgestellt. Zurückführbar ist dies auf die Natur der Untersuchung mit statischen Screenshots und der dabei fehlenden Möglichkeit zur freien Exploration. Auch hier lag die durchschnittliche Erfolgsquote bei den Fragen zur LevelTable höher (63%) als bei der Granularity Table (56%), wenn auch die Differenz nicht so stark ausgeprägt ist. Die Ergebnisse im Labor und in der Webumfrage decken sich also in ihrer leichten Tendenz zugunsten der LevelTable.

Auch wiederholt geäußertes Unverständnis gegenüber dem Konzept der Granularität als stetigem Maß für den Detailgrad weist daraufhin, dass der Versuch damit den Modalitätenwechsel zu minimieren ein Verständnisproblem aufwirft. Während das Konzept der klar voneinander abgegrenzten vier Levels der LevelTable scheinbar schnell erfasst wird, tut sich der Großteil der Testpersonen mit der Vorstellung eines stufenlos regulierbaren Parameters schwer. Es sollte daher in zukünftigen Untersuchungen überprüft werden, ob eine Rückkehr zu abgestuften Levels der Granularität bei Beibehaltung des Slider-Konzepts zu einer Verbesserung des Userverständnisses führt.

Die ausgemachten Probleme bei der GranularityTable könnten dazu verleiten, die LevelTable als die bessere Wahl für eine zukünftige Version des Metadaten-Browsers anzusehen und das Konzept der GranularityTable zu verwerfen, was jedoch eine Fehlinterpretation der Ergebnisse wäre. Es muss berücksichtigt werden, dass die Überlegenheit aus dem Vergleich von Durchschnittswerten heraus interpretiert wird, also der arithmetischen Zusammenfassung aller individuellen Testergebnisse. Dabei wird nicht berücksichtigt, dass in Einzelfällen die objektiv messbare Effizienz der Arbeit mit der GranularityTable deutlich höher war, als die mit der LevelTable. In diesen Fällen, waren auch die Sympathien vonseiten der Benutzer entsprechend auf der Seite der GranularityTable.

Es gibt offensichtlich also eine starke Abhängigkeit von der individuellen Arbeitsweise und der Erfahrung der Testperson, was auch daraus deutlich wird, dass es keine deutliche Gesamtsympa-

thie für einen der Entwürfe gibt, sondern sowohl in der Webumfrage als auch bei der Laboruntersuchung insgesamt nur eine leichte Sympathie zugunsten der LevelTable zu verzeichnen war.

Zu einer besseren Interpretation der Ergebnisse wurden in der Webumfrage Fragen zur Suchstrategie und zur Vorgehensweise bei Suchaufgaben gestellt. Grundsätzlich wurde dabei eine Unterscheidung zwischen den mit browsing-orientierten und den mit analytischen Suchstrategien arbeitenden Personen getroffen (siehe 2).

Die analytische Suche mit dem intensiven Einsatz von booleschen Ausdrücken, um eine große Treffermenge iterativ einzugrenzen, entspricht dabei sehr stark dem Use-Case, der der LevelTable zugrunde liegt. Sie ist geeignet, große Treffermengen zu visualisieren und iterativ einzuschränken, da sie die notwendigen Wechsel in den Modalitäten innerhalb eines vorgegebenen Rahmens unterstützt und auf jedem Level einen optimalen Kompromiss zwischen Detailinformation und Überblick anbietet. Gerade in der Frühphase eines Suchprozesses hat dies bei der Eingrenzung der Treffermenge große Vorteile.

Die browsing-orientierte Suche arbeitet hingegen die einzelnen Dokumente in der reduzierten Treffermenge ab, indem die Dokumente jeweils auf ihre Relevanz durch Anzeige des Inhalts geprüft werden. Diese Vorgehensweise wird durch die GranularityTable stark unterstützt, da sie dazu geeignet ist, eine eingegrenzte Treffermenge frei zu explorieren und stufenlos zwischen Modalitäten und damit auch der Metadaten- und Content-Ebene hin- und herzugleiten.

Bei der Analyse der Suchstrategie zeigten nur acht der 32 Befragten eine deutliche Orientierung zu einem der beiden Ansätze. Der Rest arbeitet offensichtlich in der Praxis mit Mischformen aus beiden Ansätzen. Von allen Befragten bekannten sich nur drei zu einem ausschließlich analytischen Ansatz, jedoch waren analytische Suchansätze trotzdem von allen Befragten stellenweise eingesetzt worden. Nur fünf der Befragten nutzen einen ausschließlich Browsing-orientierten Ansatz, jedoch nutzte nur einer von allen Befragten keinerlei Browsing-Methoden.

Es kann also keine allgemeingültige Aussage bezüglich der bevorzugten Suchstrategie in der Zielgruppe getroffen werden, außer dass die deutliche Mehrheit mit Mischformen aus beiden Ansätzen arbeitet.

Die unterschiedlichen Ausprägungen von GranularityTable und LevelTable wurden bei Betrachtung der Suchstrategie voll bestätigt: alle der ausschließlich analytisch arbeitenden Befragten bevorzugen das LevelTable Design. Von den ausschließlich browsing-orientiert arbeitenden Befragten würde nur einer die SuperTable gegenüber der Granularity Table bevorzugen.

Diese Erkenntnis spricht deutlich für eine Integration von beiden Designs in den Metadaten-Browser, da eine Mischform aus analytischer und browsing-orientierter Suche, wie sie innerhalb der Gruppe der Befragten fast ausnahmslos zum Einsatz kommt, durch die Integration beider Konzepte stark unterstützt wird und so auch die Gruppen bedient werden, die sich einer Suchstrategie verschrieben haben. Es ist anzunehmen, dass durch Einsatz beider Tables in unterschiedlichen Phasen des Suchprozesses deutlich effizienter und benutzergerechter gearbeitet werden kann. So könnte zunächst mithilfe der LevelTable die Treffermenge auf einen sinnvollen Umfang eingegrenzt werden, um anschließend in einem zweiten Schritt die enthaltenen Dokumente mithilfe der GranularityTable auf der Content-Ebene bequemer zu explorieren und zu beurteilen, ohne jedoch dabei die Metadaten-Ebene aus den Augen verlieren zu müssen.

Als grundsätzliche Bestätigung des Konzept des Metadaten-Browsers kann betrachtet werden, dass unter den Befragten der Laboruntersuchung ein generell positives Echo auf die vorgestellten Entwürfe zu verzeichnen war. Einzelne Aspekte wie die erweiterte Interaktion mit der GranularityTable über ein Kontext-Menü im Scatterplot (Abbildung 2-9: Interaktion mit Scatterplot) wurden mit emotionalen Statements wie „very clever“ oder „unusual, but I like it“ kommentiert. Auf einer Skala von 1 – 10 gaben die Benutzer beiden Designs eine durchschnittliche Wertung von 7,6 bzw. 7,2.

3.7.3 Testergebnisse für Scatterplot

Die Bedeutung des Scatterplots wurde in der Webumfrage von der deutlichen Mehrheit erkannt und richtig interpretiert (rund 65%). Jedoch wurde die Kopplung zwischen Scatterplot und Table-Visualisierung per Brushing & Linking über die selektierten Dokumente von nur ca. 40% verstanden. Diese Beobachtung deckt sich mit den Labor-Tests. Dort verstanden 75% die Bedeutung des Scatterplots aber nur 50% erkannten die Kopplung zwischen Scatterplot und SuperTable. Dies weist deutlich auf einen Mangel von visuellen Clues hin, die die Kopplung veranschaulichen und die ergänzende Nutzung von Scatterplot und SuperTable anregen. Dies könnte beispielsweise durch die automatische Fokussierung der Tabellendarstellung auf Dokumente, für die im Scatterplot ein MouseOver-Effekt aktiviert wurde, erfolgen.

In aktuellen Entwürfen des Scatterplots wurde das Glyphen-Konzept erheblich erweitert und der Scatterplot um eine Vielzahl weiterer Interaktionsmöglichkeiten (ähnlich wie in Abbildung 2-9) angereichert. Die Rolle als Visualisierung mit direktem Zugriff auf die Dokumente über Mausklick oder Bounding Box soll durch Multi-Data-Point Visualisierungen ergänzt werden, die mithilfe von Animation und Distortion Techniken die Analyse und Selektion von Clustern oder

eng beieinander liegenden Dokumenten ermöglichen. Die Möglichkeiten des visuellen Data Minings mit dem Scatterplot sollen in zukünftigen Entwürfen mithilfe von Magic Lens Mechanismen [3:Fishkin, Stone] und einem 3D Scatterplot weiter ausgebaut werden.

4 Heuristische Evaluation der INVISIP Java-Implementation

4.1 Hintergrund und Durchführung der Evaluation

Schon vor Beginn der in Kapitel 3 beschriebenen empirischen Untersuchungen wurden Teile der dort behandelten Mock-Ups als erste Komponenten einer frühen Java Applikation implementiert, die beim Abschluss des Projekts die komplette Funktionalität eines Metadaten-Browsers, wie er in Kapitel 2 und 3 noch konzeptionell bzw. prototypisch abgebildet ist, umfassen wird.

Als Ergänzung zu den in Kapitel 3 durchgeführten Maßnahmen zur Evaluation der grundlegenden Konzepte und Mock-Ups wurden die ersten fertiggestellten Java Komponenten auch einer „heuristischen Evaluation“ unterzogen. Die heuristische Evaluation oder auch „Experten-Evaluation“ basiert dabei auf der Anwendung von Usability-Heuristiken bei der Bewertung eines Systems durch mehrere Experten mit entsprechenden Fachkenntnissen.

Im Falle von INVISIP führten drei Mitarbeiter auf der Basis einer von Xerox 1995 erstellten Checkliste [4:Xerox] für heuristische Evaluation eine Beurteilung der bisher erstellten Java Komponenten durch. Diese Checkliste bildet eine Usability-Heuristik ab, die sich direkt auf die Veröffentlichungen von Nielsen et al. [4:Nielsen et al.] und Weiss [4:Weiss] bezieht und einen Leitfaden für die heuristische Evaluation in der Praxis bieten soll.

Das im Stile eines Fragebogens gehaltene Dokument umfasst 13 Themengebiete (z.B. „Visibility of System Status“ oder „Aesthetic and Minimalist Design“) die Nielsen et al. als die wichtigsten Merkmale der usability-konformen Gestaltung von Informationssystemen herausgearbeitet hat. Jedes dieser Themengebiete wird im Xerox Dokument durch eine Vielzahl von konkreten Fragestellungen an die durchführenden Experten behandelt. Beispielsweise wird im Themengebiet 4 „Consistency and Standards“ vor dem Hintergrund des oftmals übertriebenen Einsatzes von „attention-getting techniques“ nach der Verwendung von Farben im System gefragt und ob der evaluierende Experte diese für angemessen hält.

Steht eine Eigenschaft des Mock-Ups im Konflikt zur verwendeten Heuristik, wird eine nähere Beschreibung und eine Kategorisierung des Konflikts in drei Stufen (minor, major, catastrophe) verlangt.

Im Falle des Metadaten-Browsers wurde ein großer Teil der von Xerox vorgesehenen Themengebiete und auch einzelne Fragestellungen nicht berücksichtigt, da beispielsweise Fragen zu einer Kommandosprache oder Online-Hilfe auf den Metadaten-Browser (noch) nicht anwendbar waren. Trotzdem haben alle beteiligten Experten die Verwendung einer extern erstellten standardisierten Heuristik als durchaus hilfreich empfunden, da sich eine detaillierte Auflistung von verschiedensten möglichen Problemquellen beim Abprüfen der Implementation als sehr hilfreich erwies.

Für eine umfassende Bewertung wäre andernfalls eine Kenntnis und Beachtung aller potentiellen Usability-Problemquellen in Softwaresystemen notwendig gewesen. Mit dem Xerox-Fragebogen als Orientierungshilfe konnten so auch weniger erfahrene Experten Mängel erkennen, die nicht zu den üblichen Problemfeldern zählen.

Um der heuristischen Evaluation zumindest einen groben quantitativen Charakter zu geben, hat sich die einfache Kategorisierung nach Themengebiet und Schwere des Mangels bewährt. Zwar entwickeln Nielsen et al. in [4:Nielsen et al.] deutlich präzisere Instrumente zur quantitativen Auswertung der heuristischen Evaluation, dennoch schien im Falle von INVISIP die dreistufige Gewichtung der Mängel in Kombination mit den erfolgten Untersuchungen in Kapitel 3 als ausreichend, um Redesign-Vorschläge zu erarbeiten und ihnen Prioritäten zuzuweisen.

Die gewonnenen Eindrücke der drei Experten werden in Anhang C 7.3 auszugsweise aufgeführt.

4.2 Zusammenfassung und Redesign-Vorschläge

Aus den Ergebnissen der heuristischen Evaluation musste die Notwendigkeit für erhebliche weitere Entwicklungsarbeit abgeleitet werden.

Das überzeugende Grundkonzept der Schnittstelle wies in der Implementation besonders im Hinblick auf Zuverlässigkeit und Geschwindigkeit Mängel auf. An vielen Stellen wurde der Benutzer mit grafischen Artefakten, Verzögerungen und Fehlverhalten konfrontiert, was nicht nur, aber gerade auch vom Usability-Gesichtspunkt her fatal war.

Grundsätzlich musste die Implementierung dem User zukünftig einen „stabileren“ Eindruck vermitteln, als dies bisher der Fall war. Im allgemeinen wurde der Systemzustand gegenüber dem User schlecht kommuniziert. Dies reichte von elementaren Mängeln wie inakzeptablen Reaktionszeiten oder undefinierten Zuständen in der Darstellung, die beim Benutzer Wartezeiten oder die Befürchtung eines Systemabsturzes hervorriefen, bis hin zu kleinen Details wie fehlenden Indikatoren für die aktuell angewandte Sortierreihenfolge innerhalb der Tabellen.

Die Konformität der Schnittstelle mit gängigen Normen war an vielen Stellen noch unzureichend. Trotz der teilweise neuartigen Konzepte innerhalb des Userinterfaces sollte für spätere Projektphasen gewährleistet sein, dass die Verwendung von Bedienelementen und die Oberflächengestaltung (z.B. Texte, Titel, Ausrichtung) im Rahmen der üblichen Styleguides stattfindet. Hervorstechendeste Beispiele für Inkonsistenzen mit allgemein bekannten Design-Richtlinien waren die Dokumenten-Markierung mit dem grün/blauen Button und das Magic Lense Dialog Fenster (siehe Anhang C), die sich aufgrund ihrer schlechten Erwartungskonformität als besonders gravierend erwiesen.

Weiterhin wurde vorgeschlagen, die von den Experten ausgearbeiteten Konzepte zur Vereinfachung der Navigation und zur Verminderung der verwendeten Farben in Betracht zu ziehen. Es wurde bezweifelt, dass der starke Einsatz von Farben zum Highlighting wirklich eine Orientierungshilfe darstellt. Die Reduzierung der Farbzahl würde nebenbei auch die ungünstigen Farbkontraste an verschiedenen Stellen vermeiden.

Insgesamt lieferte die heuristische Evaluation zwar eine Vielzahl von Mängeln, doch schien eine Überarbeitung, bei der die geäußerten Beanstandungen entsprechend ihrer Gewichtung angegangen werden, sehr aussichtsreich. Die Verbesserung versprachen schon nach kurzer Zeit einen wesentlich überzeugenderen Gesamteindruck zu erwecken.

Ebenfalls bestätigte die heuristische Evaluation einige der Ergebnisse der Evaluationen aus Kapitel 3. In Anhang C 7.3.1 wird beispielsweise die auch im Labor und im Web kritisierte unzureichend erkennbare Kopplung zwischen SuperTable und Scatterplot erwähnt.

5 Zukünftige Möglichkeiten der Evaluation für INVISIP

5.1 Evaluation im weiteren Projektverlauf

Im Rahmen des INVISIP Projektes stellt sich nach der Durchführung der Evaluationen aus Kapitel 3 und 4 die Frage, welche Möglichkeiten der Evaluation im weiteren Projektverlauf zum Einsatz kommen sollen. Grundsätzlich stellt dabei die Methode des Usability-Testings im Labor zwar ein flexibles und mächtiges Werkzeug dar, dennoch ist damit der bereits diskutierte organisatorische und finanzielle Aufwand verbunden, der den Test im Labor im Umfang und in der Frequenz der Nutzung stark einschränkt.

Die ressourcen-schonende heuristische Evaluation stößt bei dem Versuch alle Userprobleme in späteren Phasen der Produktentwicklung vorherzusehen an ihre Grenzen, da zwischen der Wahrnehmung und der Nutzung eines System durch Experten und der Wahrnehmung und Nutzung durch reale Benutzer immer Diskrepanzen bestehen [4:Nielsen et al.]. Selbst der Einsatz von einer großen Gruppe von Experten führt daher nicht zwangsläufig zu einer vollständigen Aufdeckung aller Usability-Probleme.

Angesichts der Forderung nach geringem finanziellen und organisatorischem Aufwand ist daher erneut die webbasierte Usability-Evaluation ein Lösungsansatz. Eine weitere Userbefragung im Stile von 3.6.4 muss dabei jedoch als nicht aussagekräftig genug abgelehnt werden, da bei einem fortgeschrittenen Entwicklungsstand die grundsätzlichen Verständnisprobleme beseitigt und die grundlegenden Designkonzepte ausgereift sein sollten. Spätestens mit der Fertigstellung der ersten lauffähigen Implementation des Metadaten-Browsers als Java-Applet dominiert der Bedarf für die Untersuchung der direkten Arbeit vom User am System - ohne den Umweg über Befragung oder Mock-Ups.

Zu dieser Betrachtung sind die bisher vorgestellten Techniken der WUE aber nicht geeignet, weil sie entweder keine Aussagekraft für diesen Bereich haben (Userbefragung) oder sich das INVISIP Java-Applet nicht in ihrem Rahmen protokollieren lässt. WebQuilt kann beispielsweise nur HTTP-Zugriffe protokollieren, die einmalig beim Start des Applets auftreten. Das Geschehen im Applet an sich, wird dabei nicht protokolliert, weil es nicht über Hyperlinks und URLs erfolgt. Die bisher vorgestellten Methoden der WUE reichen dementsprechend für zukünftige Evaluationen von INVISIP nicht aus.

Generell lässt sich sagen, dass mit dem Projektfortschritt und der Annäherung an ein fertiges Produkt die Aufzeichnung und quantitative Analyse der low-level Interaktion immer mehr in den Vordergrund rückt. Erst wenn die Gestaltung und die low-level Interaktion durch Usertests des endgültigen Produkts bestätigt worden sind, werden wieder einfache Befragungen zum User-Feedback relevant. Beide Anforderungen werden durch die im folgenden vorgestellten Konzepte unterstützt.

5.2 Der webbasierte Usability-Test

Der webbasierte Usability-Test mithilfe von Remote Computing Software ist der Versuch, die Laborsituation des Usability-Tests weitgehend über das Internet abzubilden, indem es praktisch als Medium zur Verlängerung der Sinnesorgane fungiert. Wie bei einer Live-Schaltung oder Video-Konferenz wird die Anwesenheit der Testperson über Audio- und Video-Streams in das Usability Labor übertragen. Zentraler Bestandteil ist dabei der Interaktions-Kanal, der über die Remote Computing Software die „Hände“ der Testperson im Labor abbildet. Er basiert dabei auf der Vielzahl der mittlerweile verfügbaren Systeme zur kooperativen Arbeit (Microsoft Net-meeting) oder zur Fernsteuerung eines Host-Rechners durch einen Client (VNC, PC Anywhere). Ein derartiger Aufbau erlaubt somit prinzipiell auch die Tests über das Netz, die bisher nur im Labor durchgeführt werden konnten.

Wie in Abbildung 5-1 dargestellt wird dabei der Host-Rechner, auf dem die Applikation (bzw. das Test-system) installiert ist, über einen Client-Rechner gesteuert. Dazu werden alle Eingaben auf dem Client-Rechner (markiert mit I) erfasst und über TCP/IP mit einem speziellen Protokoll an den Host-Rechner geschickt. Dort werden die Daten entschlüsselt und die vorgenommenen Eingaben auf dem Host-Rechner ausgeführt. Das Feedback des Host-Rechners im Form der Veränderungen des Bildschirminhalts (markiert mit O) wird via TCP/IP zurück an den Client geschickt und dort dargestellt.

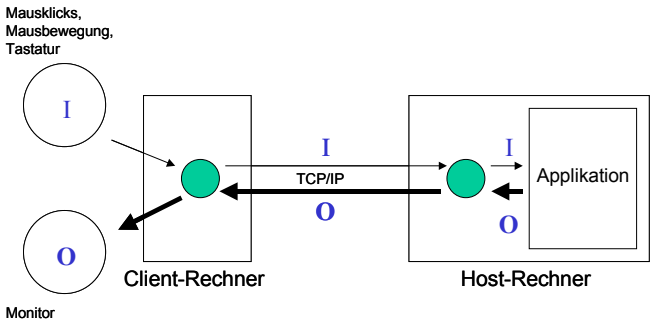

Abbildung 5-1: Schematische Darstellung Remote Computing

Dank dieser Technologie ist auf dem Client-Rechner der Desktop des Hostrechners darstellbar und dem User wird die vollständige Benutzung aller Funktionen des Hostrechners am Client ermöglicht – genügend Bandbreite wegen des umfangreichen grafischen Downstreams zum Client vorausgesetzt.

Microsoft Netmeeting bietet diese Technologie sogar zusammen mit der Möglichkeit der Internettelephonie (auch per Video), was Aufzeichnung und Moderation wie beim klassischen Usability-Test im Labor erlaubt. Das Potential dieser Technologie für das Usability Testing wurde vom OCLC Online Computer Library Center, Inc. erkannt und wird vom Usability Labor des OCLC auch für externe Projekte angeboten [6:OCLC].

Die Einrichtung eines eigenen Servers und einer entsprechenden Infrastruktur für das INVISIP Projekt erscheint durchaus als realistisch, da kein Entwicklungsaufwand notwendig ist und die notwendigen Softwarekomponenten in Microsoft Windows enthalten sind. An Hardwareausstattung wäre aufseiten der Testperson ein Mikrofon und unter Umständen eine Webcam sinnvoll, dasselbe gilt für die Seite der Testleitung, d.h. weder für Hard- noch Software müssten große Investitionen getätigt werden und die Installation dürfte schon erfahreneren Windows-Nutzern gelingen.

Leider hat diese Form des Usability Testings jedoch einige elementare Nachteile. Wie auch in [6:Gonzalez, Alvarez] kritisiert wird, werden beim OCLC Ansatz entscheidende Vorteile der WUE geopfert:

- Wird der Test durch einen menschlichen Testleiter moderiert, bleiben die Vereinbarung eines Zeitpunkts für die Testsitzung und die notwendige Anwesenheit einer Testleitung im Labor als großes Manko bestehen. Nach der Definition gilt das OCLC Verfahren daher nicht mehr als WUE.

- Die Aufzeichnung der Sitzung erfolgt nicht in einem leicht auszuwertenden formalen Rahmen, sondern es ist wie bei der Laborevaluation notwendig, große Mengen von Bild- und Tonmaterial zu sichten und zu analysieren.

- Die Installation und Konfiguration der notwendigen Software auf dem Client-Rechner ist unter Umständen sehr aufwändig, kostenintensiv und grenzt die Zahl der möglichen Testpersonen ein.

Mit dem Verzicht auf eine Moderation des Tests durch eine menschliche Testleitung könnte jedoch wieder die komplette Zeitunabhängigkeit und die automatische Abwicklung der WUE

zurückgewonnen werden. Dazu müsste das zu testende System so verändert werden, dass es bei der Interaktion einem klaren Testkonzept folgt und den Benutzer über Popup-Windows o.ä. über anstehende Aufgabenstellungen und den Testverlauf informiert. Aufzeichnung des „thinking-alouds“ oder der Mimik könnten dabei (bei entsprechender Akzeptanz durch die Testperson) auch ohne Testleitung erfolgen.

Eine weitere Verbesserung wäre die Aufzeichnung des Testverlaufs auf Ebene der Remote Computing Software (z.B. durch Speicherung der eingehenden TCP/IP Pakete). Da in der Software ja bereits Protokolle zur effizienten Speicherung von Interaktionen vorhanden sind, kann der Testverlauf dann auch nachträglich anhand der aufgezeichneten Pakete nachvollzogen, visualisiert und analysiert werden. Der Vorteil gegenüber der Aufzeichnung mit Screencam-Software liegt in den erweiterten Analysemöglichkeiten, da die Protokolle maschinell verarbeitbar und auswertbar sind. Reine Videoaufzeichnungen müssen dagegen zwangsläufig durch eine Person gesichtet werden.

Ein anderer Lösungsansatz ist das Logging von relevanten Events in der getesten Anwendung an sich, die dann im lokalen Filesystem des Hostrechners entsprechende Logfiles erstellt. Dazu ist jedoch eine speziell für das Event-Logging ausgerichtete Entwicklungsarbeit notwendig, um alle interessante Events in der Applikation mit den notwendigen Aufrufen zur Protokollierung auszustatten. Praktikabler wäre es dabei, das zu evaluierende System mithilfe eines Frameworks ohne großen Entwicklungsaufwand um entsprechende Funktionalität zu erweitern (siehe 5.3).

Die Einrichtung eines derartig erweiterten Servers zur Remote Computing WUE ist im Rahmen von INVISIP nicht umsetzbar. Für zukünftige Projekte könnte jedoch mit einer derartigen Einrichtung ein mächtiges Werkzeug für das Remote Usability Testing geschaffen werden. Die Vorteile der WUE würden so mit den Vorteilen des formalen Usability-Tests im Labor kombiniert und somit würde erstmals eine Methode für eine zeitunabhängige präzise Evaluation über das Web zur Verfügung stehen.

5.3 Das Usability-Framework für INVISIP

Eine neue Softwarekomponente, die zukünftig als flexibles Usability-Framework zur Evaluation, aber auch zum Support der post-deployment Phase künftiger Projekte dienen könnte, soll im Folgenden dargestellt werden.

Das dabei entworfene Software-Framework wird hier ausschließlich für die Verwendung mit Java-Applikationen wie z.B. INVISIP entworfen. Da sich aber objektorientierte Programmiersprachen in ihren architektonischen Möglichkeiten und in ihren APIs (z.B. zur GUI-Verwaltung) in zunehmenden Maße ähneln, ist es denkbar, das hier vorgestellte Konzept nicht nur mit Java-Applikationen und der Sun API, sondern auch mit Visual C++ und den Microsoft Foundation Classes oder Borland Delphi zu verwenden. Mit diesen Produkten wäre dann beispielsweise schon ein Großteil aller in der Praxis eingesetzten Programmiersprachen zur Entwicklung von Windows-Applikationen abgedeckt. Genauso ist die Nutzung des Frameworks auch bei Applikationen für PDAs oder für andere Mobile Devices denkbar, wo gerade Java wegen seiner Plattformunabhängigkeit häufig genutzt wird.

Grundsätzlicher Gedanke ist die Schaffung einer Klassenarchitektur zum Event-Monitoring, die das Versenden von Protokoll-Informationen an einen zentralen Protokoll-Server aus beliebigen Java-Applikationen oder Applets unterstützt. Elementarste Funktion ist das Übertragen von binär-kodierten Informationen als Datenpaket an einen zentralen Server, der über einen TCP/IP Stream mit der ausgeführten Applikation in Verbindung steht. Jeder Rechner, der Zugang zum Internet hat, kann so während des Betriebes einer Applikation - beispielsweise des INVISIP Metadaten-Browsers - über dieses Framework Daten an einen speziell zur INVISIP Evaluation eingerichteten Server senden.

Es stellt dabei keinen nennenswerten Entwicklungsaufwand dar, die Kommunikation bidirektional zu gestalten, so dass der Evaluations-Server nicht nur als Datenempfänger auftreten muss, sondern auch den Eingang von Daten beantworten oder seinerseits Informationen an den Client übermitteln kann.

Die Implementierung einer solchen Kommunikationsverbindung auf Client-Seite in Java (also aufseiten der Applikation bzw. des Frameworks) stellt dank der umfangreichen Stream- und Netzwerk-Funktionen in der Sun API keinen großen Aufwand dar.

Auf Server-Seite ist aufgrund eines möglicherweise hohen Datenaufkommens und simultanen Connections zu einer Vielzahl von Clients ein umfangreicheres Gegenstück an Server-Software notwendig, das sich aber zweifellos ebenfalls vollständig in Java realisieren lässt. Hinweis darauf sind die erfolgreichen Umsetzungen von Web-Server- und HTTP-Proxy-Diensten in Java, wie sie z.B. auch bei WebQuilt zum Einsatz kommen. Sollten sich dennoch in der Praxis Probleme auf Server-Seite wegen mangelnder Performance o.ä. zeigen, ist dort auch andere Technologie einsetzbar, da die Kommunikation über das standardisierte TCP/IP erfolgt. TCP/IP wird

heutzutage von allen gängigen Entwicklungsplattformen unterstützt. Die Server-Seite ist also nicht an die Beschränkungen von Java gebunden.

Grundgedanke des Usabilty-Frameworks ist es, dass die Applikation vom ausführenden Client aus Bericht über alle usability-relevanten Ereignisse während der Nutzung erstattet. Was dabei als „relevant“ bezeichnet wird, kann durch eine Kontaktaufnahme zwischen Client und Server im Vorfeld ausgehandelt werden.

Derartige Ereignisse können beispielsweise nur der Start oder das Ende einer Usersession sein, was jedoch schon ausreichend Informationen für eine Untersuchung der Nutzungshäufigkeit darstellt. Auf der Server-Seite können durch Erfassung der einzelnen Usersessions und ihrer Dauer bereits Statistiken über die Anzahl und die geografische Verteilung der Nutzer (soweit aus der IP-Adresse ersichtlich), sowie über durchschnittliche Arbeitszeiten mit der Software erstellt werden.

Je genauer über die Einzelereignisse während des Betriebes berichtet wird, desto umfassender ist das Bild von der Art und Weise der Nutzung. Wenn beispielsweise das Öffnen von Fenstern in der Applikation protokolliert wird, können Informationen über typische Workflows und Vorgehensweisen gesammelt werden. In Kombination mit der Dimension Zeit können so für Software-Applikationen umfangreiche Daten über Verweildauern und Navigationspfade erfasst werden, wie es bisher außerhalb des Labors nicht möglich war.

Auf der höchsten Stufe der Protokollierung werden dann alle Events (Tastatureingaben, Mausbewegungen, Selektionen etc.) übermittelt, so dass eine Usersession in allen Details verfolgt, nachvollzogen und analysiert werden kann. Die Qualität der so gewonnenen Daten steht denen des Usability-Tests im Labor in nichts nach, ausgenommen des „thinking-alouds“ und der Videoaufzeichnung der Testperson. Eine derartige AV-Aufzeichnung wäre zwar über Mikrofone oder Webcams technisch durchaus realisierbar, würde in der Praxis jedoch sicherlich auf wenig Akzeptanz stoßen. Eine weitere Verbreitung von IP-Bild-Telephonie könnte jedoch in Zukunft auch diese Möglichkeit eröffnen.

Ein Vorteil des dargestellten Protokollmechanismus liegt in der großen Flexibilität. Die Anwendung kann bei Bedarf (z.B. durch ein Switch in der Kommandozeile) in den Protokollmodus geschaltet werden, der sich vom normalen Betrieb nur durch das unsichtbare Versenden der Informationen an den Protokollserver über einen TCP/IP Port unterscheidet.

Der Umfang der gesendeten Information lässt sich über fertig konfigurierte Loglevels des Frameworks bestimmen, die mit höherem Grad zunehmend detaillierte Informationen an den Protokollserver liefern. Dabei kann die Wahl des Loglevels auch in Abhängigkeit von den vorhandenen Systemressourcen und der Bandbreite der Internet-Verbindung stattfinden, um trotz Event-Logging einen optimalen Betrieb zu gewährleisten. Weiterhin können auch eigene Loglevels definiert werden, um speziellen Produktanforderungen oder Szenarien gerecht zu werden.

Das Framework kann sowohl bei Java Mock-Ups als auch bei kompletten Applikationen zum Einsatz kommen. Während es in fertigen Applikationen eher zu Nutzerstatistiken oder zum Quality Feedback verwendet wird, kann es genauso auch zur Protokollierung von Testaufgaben dienen. Sinnvoll wäre dazu ein Testmodus, der einen einheitlichen moderierten Testablauf und die dafür nötigen Einstellungen und Informationsfenster steuert (siehe 5.3.1).

Wenn Protokollierung oder Testmodus vom Benutzer nicht erwünscht ist, werden sie in der Kommandozeile oder in den Einstellungen der Applikation deaktiviert. Natürlich könnte eine Applikation ohne Zustimmung des Benutzers generell immer versuchen, Kontakt zu einem Protokollserver aufzunehmen. In diesem Fall sollten jedoch Probleme des Datenschutzes berücksichtigt werden (siehe 5.3.1).

Soll die Applikation nicht auf Client-Seite, sondern per Remote Computing (siehe 5.2) auf dem Server direkt ausgeführt werden, ist anstatt des Versendens der Protokollinformation über das Internet auch eine Protokollierung der Daten nach „localhost“ oder in ein lokales File möglich. Auf diese Weise können Software-Prototypen - ohne eine Distribution an die Testgruppe – ausschließlich auf dem Protokollserver/Host-Rechner zur Evaluation über das Web angeboten und ausgeführt werden.

5.3.1 Grundprobleme des Usability-Frameworks

Wie oben erwähnt sind bei einer Umsetzung des Usability-Frameworks einige Probleme zu diskutieren, die im Folgenden kurz angesprochen werden sollen.

- **Internet-Anbindung:** Die Protokollierung kann nur bei einer ausreichend leistungsfähigen Internet-Verbindung verwendet werden. Sofern kein bidirektionaler Austausch von Daten zwischen Client-Applikation und Server vorgesehen ist, kann auch eine Protokollierung in ein lokales Logfile als Puffer für einen späteren Versand dienen. Dies schränkt den Nutzen aber stark ein.

- **Firewalls/Router:** Immer mehr Unternehmen und Benutzer schützen ihre LANs oder PCs vor dem Empfangen und Versenden von ungewünschten Daten aus oder zum Internet mit Hardware- oder Software-Firewalls. Diese blockieren üblicherweise nicht genutzte TCP/IP Ports. Solche Ports müssen aber zur Kommunikation mit dem Protokollserver verwendet werden. Auch häufig eingesetzte Techniken zur gemeinsamen Nutzung eines Breitbandzugangs mit Routern (IP-Masquerading oder NAT) können zu Verbindungsproblemen führen. Hier ist es notwendig, Erfahrungen aus der Entwicklung von anderen Internet-Diensten (z.B. Instant Messaging, File Sharing) heranzuziehen.

- **Datenschutz:** Ein sehr wichtiger Aspekt ist die allgemeine Skepsis gegenüber einer Software mit direktem Kontakt zum Entwickler/Hersteller über das Internet. Sogenannte Spyware oder auch Komponenten von Microsoft Windows zum „Ausspähen“ installierter Software oder verwendeter Produkte genießen innerhalb der Benutzergemeinde einen außerordentlich schlechten Ruf. Oftmals ist es die Intransparenz der dahinterstehenden Kommunikationsvorgänge, die aufseiten der Benutzer zu großem Misstrauen und geringer Akzeptanz gegenüber der Verwendung solcher Software führt. Selbst überschaubare Vorgänge, wie das Absenden von Bug-Reports durch Feedback Agents in Windows XP oder Netscape, werden kritisch begutachtet und als oft unerwünscht empfunden.

  Es ist daher nötig, diese Thematik offensiv zu diskutieren, die Kommunikationsvorgänge transparent darzustellen und damit eine entsprechende Vertrauensbasis bei Testpersonen bzw. Endkunden zu erreichen.

- **Moderation:** Die komplette Aufzeichnung der Interaktion mit einer Applikation kann nur Grundlage für eine dem Usability-Test ähnliche Evaluation sein. Unverzichtbar ist die Wahrung eines formalen und inhaltlichen Rahmens für den normalerweise die Testleitung verantwortlich ist. Bei der Verwendung des Usability-Frameworks ist diese nicht vorhanden, was gleichzeitig Vorteil (siehe 3.6.1, Kostenersparnis, geringer Organisationsaufwand) und Nachteil darstellt. Die fehlende Moderation muss innerhalb der Applikation durch Informationsfenster, Beispieldaten und durch die Wahrung von Zeitlimits und klaren Abfolgen in den Arbeitsschritten ersetzt werden. Es muss also eine Art „Fernsteuerung“ der Applikation durch den Protokollserver oder das Framework möglich sein. Für die Umsetzung wäre die Integration einer einfachen Skriptsprache innerhalb des Frameworks denkbar. Dennoch bleibt die Frage nach einer Lösung ohne einen derartigen Entwicklungsaufwand zu klären.

- **Integration:** Um das Framework in der oben beschriebenen Weise einzusetzen, müssen eine Vielzahl von Ereignissen (in einigen Fällen sogar alle Ereignisse, die mit Eingabe oder GUI zusammenhängen) von der Applikation durch Aufruf des Frameworks an den Protokollserver weitergemeldet werden. Während die Integration dieser Aufrufe bei neuentwickelten Applikationen wenig aufwändig erscheint, muss im Falle einer nachträglichen Integration mit erheblichen Schwierigkeiten gerechnet werden. Abhilfe schaffen hier die OO-Eigenschaften von Java und dessen API, die durch Vererbung und das EventListener-Konzept eine Integration erheblich vereinfachen (siehe 5.3.2).

Angesichts dieser Probleme und des hohen Entwicklungsaufwands für das Framework erscheint eine Realisierung im Rahmen von INVISIP als unmöglich. Als eigenständiges Produkt wäre das Framework dennoch lohnenswert, da seine grundlegende Architektur völlig isoliert von der konkreten Java-Applikation ist und es damit für zukünftige Java-Applikationen aller Art zur Verfügung stehen könnte. Mittel- bis Langfristig wäre ein derartiges Framework dazu geeignet, Bestandteil der Sun API Klassen zu werden, um allen größeren Java-Applikationen die Möglichkeiten der WUE und des Quality Feedbacks zugänglich zu machen.

5.3.2 Architektur des Usability-Frameworks für Java

In Java werden alle eingehenden relevanten Events wie Tastatureingaben, Mausbewegungen, Fensterveränderungen etc. über sogenannte EventListener an die Applikation weitergegeben. Die von der Java API standardmäßig angebotenen Listener sind dabei jeweils speziell für einen Aufgabenbereich implementierte Klassen, deren Methoden dann aufgerufen werden, sobald die für sie relevanten Events eingetreten sind. Sollen z.B. in einem Fenster Tastatureingaben möglich sein, so wird dieses von der Applikation mit einem KeyListener versehen, der vom System aufgerufen wird, sobald Keyboard-Events verzeichnet werden.

Um aber mit dem Framework die Events dieses Listeners zu protokollieren, muss dieser mit dem Aufruf der Protokollfunktion des Frameworks ergänzt werden. Damit dies ohne Veränderung des Codes der Applikation erfolgen kann, wird dazu ein neuer KeyListener2 über Vererbung mit den Funktionen des ursprünglichen KeyListener versehen und zusätzlich um die Aufrufe der Protokollfunktionen des Frameworks erweitert. Der so erweiterte, neue KeyListener2 ersetzt dann den ursprünglich verwendeten KeyListener in der Applikation.

In der Praxis heißt dies, dass eine Integration des Frameworks prinzipiell schon durch simple Suchen/Ersetzen-Operationen im Sourcecode möglich wird. Die in der Applikation verwendeten

Listenerklassen können einfach durch den Namen ihrer erweiterten Erben aus dem Framework ersetzt werden (ersetze „KeyListener“ durch „KeyListener2“). Die Architektur oder der Code der Applikation müssen dabei nicht grundsätzlich verändert werden. Weiterhin werden die EventListener oft in der ganzen Applikation gleich verwendet, so dass durch simples Ersetzen eines KeyListeners alle Tastatureingaben in der Applikation protokollierbar werden.

Um bei sehr umfangreichen Logging (z.B. präzises Logging der Mausbewegungen) eine Überflutung des TCP/IP Streams mit kleinen Datenpaketen zu vermeiden, könnten Pufferklassen zum Einsatz kommen, die je nach Priorität des Ereignisses die eingehenden Daten zunächst sammeln und erst bei ausreichender Datenmenge oder geringer Netzlast absenden. Da die Speicherung im Puffer schon zum Eingangszeitpunkt jeweils mit einem Timestamp versehen wird, ist eine entsprechende Rekonstruktion der Abfolge der Ereignisse nach dem Absenden der Pakete auf dem Protokollserver keine Schwierigkeit.

5.3.3 Quality Feedback und weitere Dienste im Usability-Framework

Die so geschaffene Funktionalität kann nicht nur zum Event-Monitoring genutzt werden. User-Feedback oder kritische Usability-Issues können auf Knopfdruck und ohne Umweg über E-Mail direkt mithilfe eines Report-Formulars an den Server weitergesendet werden. Weitere hilfreiche Informationen für die Entwickler (z.B. den Dump wichtiger Variablen) werden dabei automatisch hinzugefügt. So bleibt auch nach dem Release in der post-deployment Phase ein kontinuierlicher Fluss von Feedback zur Verbesserung des Produkts erhalten. Die Möglichkeiten eines solchen direkten Feedbacks speziell im Usability-Kontext werden in [6:Hartson, Castillo] näher beschrieben.

Die aus Windows XP oder dem Netscape Navigator bekannte Quality Feedback Agents sind in der Lage automatisch auf Abstürze oder undefinierte Zustände zu reagieren, um beim Auftreten von Fehlern, notwendige Informationen an die Entwickler abzusenden. Dies ist insbesondere sinnvoll, falls Inkompatibilitäten zu anderen Produkten vorliegen oder Systemabstürze unter Laborbedingungen nicht reproduzierbar sind. Diese automatischen Bugreports müssen dabei allerdings vom Framework selber ausgelöst werden. Wie bei den EventListenern könnte sich hier das Konzept der Java-Exceptions als äußert hilfreich erweisen, um derartige Ausnahmezustände ohne weitgehende Änderungen im Code der Applikation über das Framework zu erfassen und entsprechend zu bearbeiten.

Ebenfalls attraktiv könnte eine Nutzung des Rückkanals vom Protokollserver zur Distribution von Updates oder Patches sein. Es wäre denkbar, dabei eine ähnliche Funktionalität anzubieten, wie es die Microsoft Windows Update Seite bietet. Abhängig von der vorliegenden Version der Applikation, werden sinnvolle oder dringende Updates und Patches vorgeschlagen und per Download direkt auf den Client übertragen. Eine automatische Installation könnte durch Austausch entsprechender Java-Class- oder Jar-Files erfolgen.

5.3.4 Datenformate und Visualisierungen für das Usability-Framework

Das National Institute for Standards and Technology der USA hat mit FLUD [6:Cugini, Laskowski] ein Framework für das Logging von Usability-Daten geschaffen, dessen standardisiertes Fileformat und dessen Parser als Komponenten des hier vorgestellten Usability-Frameworks dienen könnten.

FLUD legt dabei den Schwerpunkt auf die Schaffung eines universellen Austauschformats für die Aufzeichnung von Interaktion mit Web-Applikationen, das die Basis für einen zukünftigen Standard für Usability-Werkzeuge legen soll. Das NIST hat im WebMetrics Paket schon verschiedene Tools zur Aufzeichnung, Wiedergabe und Visualisierung von FLUD-Files veröffentlicht.

Im allgemeinen sind aber fast alle Werkzeuge und Veröffentlichungen, die sich mit der Aufzeichnung und Visualisierung von Usability-Logging Daten beschäftigen, nur auf die Interaktion mit Webseiten beschränkt. Der Nutzen dieser Technologien für das Usability-Engineering könnte jedoch erheblich erweitert werden, wenn sie nicht nur im begrenzten Rahmen der Web-Usability zum Einsatz kommen würden.

Betrachtet man die in FLUD umgesetzten Grundkonzepte, so ist es denkbar, den vorhandenen Standard zu nutzen, um mit ihm auch die Interaktion mit Applikationen abzubilden. Sollten dennoch Lücken in den Möglichkeiten der Protokollierung bleiben, könnten spezielle Einträge in den FLUD-Files, die als Kommentare überlesen werden, mit diesen spezifischen Informationen versehen werden, damit die Daten dennoch abwärtskompatibel zu den bisher vorhandenen Tools bleiben. In neuen Werkzeugen könnten diese Daten zukünftig berücksichtigt werden.

5.3.5 Intelligente Software-Agenten für das Usability-Framework

Sowohl in [6:Hilbert et al. 1998] und [6:Hilbert et al. 1999], als auch in [6:Gonzalez, Alvarez] werden Agenten-Systeme vorgestellt, die als mögliche Alternative oder zur Weiterentwicklung

des hier beschriebenen Konzepts dienen könnten. Dabei kommen auf Client- bzw. Applikations-Seite Agentensysteme zum Einsatz, um die Events aufzuzeichnen, zu verarbeiten und sie an den Protokollserver weiterzuleiten.

Der Vorteil dabei soll vor allem in der komplexen Vorverarbeitung der Informationen für den Server durch die Agenten und in ihrer leichten Integration liegen. Bei dem von Gonzalez et al. entwickelten ANTS System erfolgt dies beispielsweise durch kleine autonome Java-Applets, die auf Webseiten platziert werden, um dort z.B. die Mausbewegung und die einzelnen Interaktionsschritte aufzuzeichnen. Nach der Übermittlung an den Protokollserver stehen so nicht nur einzelne Seitenwechsel wie beim Weblogging, sondern auch die intra-page Events zur Analyse zur Verfügung.

Während ANTS als durchaus sinnvolle Weiterentwicklung des Weblogging für die Web-Usability betrachtet werden kann, ist sein Nutzen für das Event-Monitoring in Applikationen fraglich. Das was in ANTS als Agent bezeichnet wird, leistet in seiner Funktionalität nicht mehr, als die hier beschriebenen erweiterten Listener mit Aufrufen in das Usability-Framework. Die weiteren Dienste des hier beschriebenen Frameworks werden jedoch nicht angeboten. Die ANTS Architektur ist weiterhin wegen ihrer Ausrichtung auf Java Applets als Agenten nur für die Integration in Websites geeignet. Es könnten daher allenfalls sehr grundlegende Konzepte in den Rahmen des Usability-Frameworks übernommen werden.

In [6:Hilbert et al. 1998] und [6:Hilbert et al. 1999] wird ebenfalls ein Framework zum Event-Monitoring entworfen, wobei dort Agenten-Systeme in die Lage versetzt werden sollen, die low-level Interaktionsschritte zu sammeln, um anhand dieser das Userverhalten zu interpretieren. Dabei werden durch die Agenten bestimmte Erwartungen an den weiteren Verlauf der Interaktion entwickelt. Erst wenn die Agenten feststellen, dass die tatsächliche Handlungsweise des Benutzers ihrer Erwartungshaltung widerspricht, werden diese Widersprüche formuliert und an den Protokollserver gemeldet.

Dieses System des Expectation-driven event monitoring (EDEM) wird damit begründet, dass ein Event-Monitoring einer großen Menge von low-level Information von einer Vielzahl von Arbeitsplätzen zur Verarbeitung an einer zentralen Stelle zu umfangreich sei. Eine Reduktion der anfallenden Daten auf kritische Events durch den EDEM Mechanismus wird von Hilbert et al. als notwendig für die Verwendung von Event-Logging in größerem Rahmen bezeichnet.

Das EDEM Konzept erscheint für eine Weiterentwicklung des Frameworks attraktiv, jedoch sollten Entwurf und Möglichkeiten der Agenten im Vorfeld ausführlich geprüft und diskutiert

werden. EDEM verlangt, dass die low-level Interaktion durch Beobachtung durch die Agenten analysiert und mit erwarteten Verhaltensweisen verglichen wird. Dieser Transfer der einzelnen low-level Events auf eine semantische Ebene, die das Userverhalten „verstehen“ oder „voraussagen“ soll, erscheint schwer in der Praxis umsetzbar. Eine ähnliche Interpretation zur Ermittlung des mentalen Modells beim Usability-Test im Labor erfordert dort die Auswertung und die Expertise eines Fachmanns. Nicht zuletzt dazu werden im Labor auch „thinking-aloud“ und Mimik der Testperson hinzugezogen.

Die Entwicklung von Agenten zur richtigen Interpretation erscheint daher sehr aufwändig und arbeitsintensiv. Ob die reduzierte und vorverarbeitete Datenmenge durch EDEM diesen großen Entwicklungsaufwand gegenüber simplen Event Logging rechtfertigt, sollte im Vorfeld untersucht werden.

6 Zusammenfassung

Auf der Basis der Erfahrung im INVISIP Projekt wird in 6.1 eine Gegenüberstellung der diskutierten Evaluations-Methoden vorgenommen. Behandelte Kernfragen, Vor- und Nachteile, geeignete Zeitpunkte zum Einsatz und Aufwand werden dort in einer Tabelle zusammengefasst. Das Ziel ist es nicht, eine umfassende Typologie und Darstellung aller gängigen Usability-Methoden zu liefern, sondern nur auf die Methoden einzugehen, die sich im Rahmen von INVISIP als praxis-relevant und zielführend erwiesen haben oder deren Anwendung diskutiert wurde.

Die hier vorgenommene Klassifizierung entspricht dabei nicht den üblichen Abgrenzungen der Methoden in der Literatur. Die INVISIP Praxis zeigte, dass gerade der Einsatz von Mischformen verschiedener Methoden (siehe Webbefragung 3.6.4) und die ergänzende Durchführung mehrerer Evaluationen den Nutzen im Verhältnis zum Aufwand wesentlich erhöhen können. Solche Mischformen werden in der Tabelle trotzdem als eine Methode wegen der gemeinsamen Zielsetzung und Durchführung aufgefasst.

Generell zeigt die Erfahrung bei INVISIP, dass es keine „ideale“ Evaluationstechnik in einer Projektphase gibt. Individuelle Projektaspekte beeinflussen den Nutzen einer Evaluation fortwährend. Finanzielle und organisatorische Aspekte können die Wahlmöglichkeiten erheblich einschränken. Verschiedene Vorgehensmodelle des Software- oder Usability-Engineerings bedingen dabei unterschiedliche Abfolgen von Projektphasen und damit Evaluationen an unterschiedlichsten Stellen.

Eine wertvolle Usability-Evaluation zeichnet sich in erster Linie durch eine im Aufwand angemessene und ausreichend valide Beantwortung der akuten Fragen an die Designkonzepte, Prototypen oder das fertige Produkt aus. Ihr konkreter Nutzen für das Projekt liegt in der Identifizierung von Usability Problemen in den Designkonzepten und in der Entwicklung von Redesignvorschlägen. Daher sollten Kosten und Nutzen immer im Verhältnis stehen. Die präzise Umsetzung von Testmethoden oder Evaluationstechniken aus der Literatur oder „Userbeteiligung um jeden Preis“ ist für das Projekt kein Wert an sich.

Im Sinne der von Nielsen propagierten „Guerrilla HCI“ [6:Nielsen 3] soll hier eine Flexibilisierung im Einsatz von Vorgehensmodellen und traditionellen Methoden der Usability Evaluation angeregt werden. Dabei erheben die hier vorgestellten neuen Ansätze und Mischformen nicht

den Anspruch, in Methodik und Validität den präzisen klassischen Tests zu entsprechen, erreichen jedoch mit verhältnismäßig geringem Aufwand einen großen Nutzen für das Projekt, wie anhand von INVISIP demonstriert wird.

Die hier präsentierten Werkzeuge können Nielsens Ansatz des „Discount Usability Engineering“ unterstützen, da sie helfen, die Hemmschwellen bei der Nutzung des Usability Engineerings (komplizierte und kostenintensive Evaluationstechniken, hohe Investitionen) zu senken. Insbesondere die webbasierten Evaluationstechniken sind geeignet, auch kleinen Projekten ohne Zugriff auf Usability Labore die Methoden des Usability Testings an die Hand zu geben.

Generell können die hier dargestellten Methoden bei richtiger Anwendung ein Verhältnis zwischen Kosten und Nutzen erreichen, dass es auch kleinen Unternehmen oder Entwicklergruppen ermöglichen sollte, zukünftig Usability-Aspekte bei der Produktentwicklung zu berücksichtigen. Somit können diese Methoden einen wertvollen Beitrag zur Verbesserung der Qualität der uns zur Verfügung stehenden Informationssysteme leisten.

6.1 Tabellarische Gegenüberstellung der Evaluations-Methoden

Im Folgenden sind die diskutierten Evaluations-Methoden tabellarisch gegenübergestellt. In der Spalte Durchführung und Auswertung befindet sich eine Bewertung des Aufwands zur Durchführung/Gestaltung/Organisation bzw. zur Auswertung für die jeweilige Methode (Skala: „--“: sehr großer Aufwand bis „++“: sehr geringer Aufwand).

Methode	Kapitel	Hardware, Software, Personal	Themen (primär)	Themen (sekundär)	Durchführung	Auswertung
Userumfrage (Print)	(3.6)	Papier, Postweg	Zielgruppenanalyse, Benutzerprofile, IT-Erfahrung, bisherige Arbeitsweise, typische Arbeitsgänge	Verständnisgrad für abstrakte Konzepte im Design, hypothetische Arbeitsabläufe, gewünschte/erwartete Interaktion	- -	0
Userumfrage (Web)	3.6	Webserver, Skripte, WWW	Zielgruppenanalyse, Benutzerprofile, IT-Erfahrung, bisherige Arbeitsweise, typische Arbeitsgänge	Verständnisgrad für abstrakte Konzepte im Design, hypothetische Arbeitsabläufe, gewünschte/erwartete Interaktion	0	+

Usertest (Labor)	3.5	Usability-Labor, Aufzeichnung Audio/Video, System oder Mock-Up auf Testrechner Testleitung, Protokollant	Verständnisgrad für abstrakte Konzepte im Design, visuelle Gestaltung, low-level Interaktion, mentales Modell der Person, Mistakes, Slips, Effektivität/Effizienz der Arbeitsabläufe *prinzipiell alle Aspekte!*	Zielgruppenanalyse, Benutzerprofile, IT-Erfahrung, bisherige Arbeitsweise, typische Arbeitsgänge über Pre-Test/Post-Test	- -	- -
Usertest (Weblogging)	3.5/3.6.3	Webserver, WebQuilt, Website oder Mock-Up im Web	Verständnisgrad für abstrakte Konzepte im Design, Navigationsstrukturen, Mistakes, mentales Modell von der Site-Struktur/Vorgängen, Effektivität/Effizienz der Arbeitsabläufe *aber keine intra-page Aktivitäten/low-level Interaktion!*		+	+
Usertest (Web, OCLC)	3.5/5.2	Usability-Server mit System oder Mock-Up via Netmeeting, Client-Rechner mit Netmeeting, Webcam, Mikrofon Testleitung, Protokollant	Verständnisgrad für abstrakte Konzepte im Design, visuelle Gestaltung, low-level Interaktion, mentales Modell der Person, Mistakes, Slips, Effektivität/Effizienz der Arbeitsabläufe *prinzipiell alle Aspekte!*		-	-
Usertest (Web, Usability-Framework)	3.5/5.3	System oder Mock-Up bei Testperson mit Internetverbindung, Protokollserver im Web	Verständnisgrad für abstrakte Konzepte im Design, visuelle Gestaltung, low-level Interaktion, mentales Modell der Person, Mistakes, Slips, Effektivität/Effizienz der Arbeitsabläufe *prinzipiell alle Aspekte!*		++	+
Nutzungsdaten (Usability-Framework	3.5/5.3	System oder Mock-Up bei Testperson mit Internetverbindung, Protokollserver im Web	post-deployment Phase: Nutzungsdauer, Nutzungshäufigkeit, geografische Verteilung der Benutzer		++	++
Userreport (Usability-Framework)	3.5/5.3.3	System oder Mock-Up bei Testperson mit Internetverbindung, Protokollserver im Web	post-deployment Phase: Abstürze, Systemfehler, Exceptions vom User gemeldete Usability Issues oder Verbesserungsvorschläge		++	++

7 Anhang

7.1 Anhang A: Laborevaluation

Im Folgenden finden sich die Aufgabenstellungen für die Laborevaluation der Mock-Ups für LevelTable und GranularityTable.

In **Fettdruck** wird der aktuelle Zustand des Mock-Ups nach/vor dem Bearbeiten einer Aufgabe angegeben. Der eingerückte Text gibt die Aufgabenstellung wieder, die der Testperson gestellt worden ist. In *Kursivdruck* werden interne Vermerke für die Testleitung bezüglich der zu klärenden Fragen und der dafür zur Verfügung stehenden Zeit dargestellt.

7.1.1 Aufgabenstellung LevelTable

Level 1 LevelTable Vollansicht

Imagine you just made a search, and now you can see the result screen. Please take a moment, and then describe what you see.

2 min. Goal: First impression

You want to have a little bit more detailed view . Please find a way to get it.

1 min. Goal: Click in document or click on Level Button

Please go to Level 2

1 min. Goal: Make familiar with Level concept Find Level Button and click.

Level 2 LevelTable Vollansicht

Please describe what has changed compared to level 1

1 min. Goal: Get familiar with new view.

How would you change the order of the colums, e.g. put “gis” to the second column?

30 Sec. Goal: Drag and drop, familiarize with possibilities

Now look at the results more thoroughly. Please find a result which is a document from an university.

1,5 min. Goal: Find Document Type text/academic. Naming, visual clearness

You want to work with this document. Please find a way to mark it.

1 min. Goal: Checkbox, click on checkbox.

The INVISIP system also works with a graphical search tool, named the scatterplot. Please find it!

1 min. Goal: Find button Scatterplot upper left corner.

Level 2 LevelTable + Scatterplot

In the Scatterplot, please click on the most important document.

1 min. Goal: Find document with highest relevance. Is relevance clear to everybody?

Please describe what happened. Can you think of a name for what happened?

2 min. Goal: Describe interaction Scatterplot – LevelTable. Does the naming make sense to everybody?

Please go to level 3

30 sec. Goal: Is the level concept clear by now?

Level 3 LevelTable + Scatterplot

Please remove the scatterplot

30 sec. Goal: recall the position of button Scatterplot

Now, please find the most unimportant document.

1 min. Goal: Check values of relevance bars.

After you found it, please move on to level 4

30 sec. Goal: repeat action, no goal.

Level 4 LevelTable

Please describe what you can see.

2 min. Goal: familiarize with LevelTable

Now click on the first document. What happens?

1 min. Goal: Describe

Please name some differences between “Title” and “Abstract”

2 min. Goal: Wording, are categories clear?

Now please get the Scatterplot and describe what you see.

1 min. Goal: Find Browser, describe Browser

7.1.2 Aufgabenstellung GranularityTable

Level 1 GranularityTable

This is the Granularity Supertable. Please take a moment and describe briefly, what you see.

1 min. Goal: First Impression

Imagine you just made a search, and now you can see the result screen. Please take a moment, and then describe what you see.

2 min. Goal: First impression, describe GranularityTable

The INVISIP system also works with a graphical search tool, named the scatterplot. Please find it!

1 min. Goal: Find Scatterplot-Button.

Now take a look at the whole screen, and briefly describe what you see.

1 min. Goal: Impression and description of whole system

In the scatterplot, please click on the most important document.

1 min. Goal: Find document with highest relevance. Is relevance clear to everybody?

What do you think does granularity mean?

1 min. Goal: Granularity Table understood?

To find out, please click left of the slider below of granularity.

1 min. Goal: Visual structure clear. Monitor could help here.

Level 2 GranularityTable + Scatterplot

Now we are in Level 2. What has changed?

2 min. Goal: first impression and description

For your search, you think the term “geo” is the most relevant. What can you do?

2 min. Goal: find sort functionality on rows. Click on “geo”

Now, please go to the next level

1 min. Goal: Granularity Concept understood? Use of overall slider.

Level 3 GranularityTable + Scatterplot

The text column has gotten bigger. Can you imagine, why?

2 min. Goal: Explore possibilities of granularity concept

Please go to the next level

Monitor can help, if slider not found.

Level 4 GranularityTable + Scatterplot

You suspect that the text in the upper row is interesting. Please find a way to read it in whole

2 min. Goal: document granularity slider.

Please go to the next level

30 sec. No Goal.

Level 5 GranularityTable + Scatterplot

Please describe briefly what has changed

2 min. Goal: Describe visualization of text blocks and keyword highlighting. Purpose: navigation

What do you think does the black arrow mean?

1 min. Goal: understand navigational concept

What do you suppose do the coloured squares mean?

1 min. Goal: understand the text segmentation concept.

Please go to the next level

30 sec. No Goal

Level 6 GranularityTable + Scatterplot

Please describe briefly what has changed

1 min. Goal: describe Browser

You have looked exactly for this document and you are happy that you could find it. In a real search, would you be satisfied with the way you could find the document?

2 min. Goal: retrospective satisfaction.

Would you have used the scatterplot also?

2 min. Goal: opinion from participant.

Which visualization would you prefer for your daily work, the Scatterplot or the GranularityTable?

2 min: Goal: opinion from user.

7.1.3 Beispielauswertung einer Testsitzung

Tasks:

1. ist verwirrt von Level vs Granularity Slider („ist Granularity stetig?“) kein Verständnis der Oberfläche, kein Verständnis des Begriffs „Relevance“, Versteht Bedeutung der Spalten nicht
2. erkennt die Zunahme von Informationen für gewähltes Dokument und kann einzelne Spalten jetzt interpretieren
3. versteht Zunahme von Information durch Level 2, kann alle Spalten interpretieren
4. versteht Achsenbezeichnung und stellt selbständig Beziehung zwischen Supertable und Scatterplot,
5. NA
6. findet Dokument sofort, erkennt Beziehung zwischen Supertable und Scatterplot,
7. NA
8. “Relevance Curve” versteht Bedeutung nicht, erwartet Beziehung zu Scatterplot
9. NA
10. versteht Bedeutung von „Abstract“
11. “Stacked columns” wird nicht verstanden, verwirrt durch Browser Fenster, vermutet Suchmaschine in Browser
12. versteht Konzept Fokus-Selektion

Kategorie und Bewertung (-,0,+)	
Verständnis Super Table/Result Table, Bedeutung der Spalten	0
Verständnis für Keywords, Keyword Highlighting und Relevanzen	-
Bedeutung des Scatterplots	+
Interaktion mit Scatterplot, Beziehung zwischen Scatterplot und Tabelle, Tooltips, Zooming	+
Verständnis für Granularitäts/Level-Konzept	0
Interaktion mit Tabelle (Sortierung, Markierung...)	+
Visualisierung/Navigation im Text (Stacked Columns, Relevance Curve, …)	-

Summary:

- „Stacked columns“, „Relevance Curve“ wird nicht verstanden, Beziehung zwischen Farben und Suchbegriff darin wird nicht verstanden
- kann Spaltenbedeutung erst bei höherem Level erkennen
- ist verwirrt von Gegensatz Level (ordinal) Granularity (stetig?)
- verwirrt durch Browser Fenster, vermutet Suchmaschine in Browser

7.2 Anhang B: Webbasierte Evaluation

Da eine Printversion der kompletten Webuntersuchung als Hardcopy den Umfang dieser Arbeit sprengen würden, wird in 3.6.6 exemplarisch eine der Fragestellungen präsentiert.

Zugriff auf die kompletten Untersuchung kann unter folgender URL erhalten werden:

http://merkur25.inf.uni-konstanz.de/jetter

Zur Authentifizierung ist die Eingabe von „studie“ als Username und „ukon“ als Passwort notwendig.

7.3 Anhang C: Heuristische Evaluation

Im Folgenden sind zwei Beurteilungen zu den Kategorien „Visibility of System Status“ und „Consistency and Standards“ exemplarisch aufgeführt. Beide Kategorien wurden ausgewählt, da in ihrem Rahmen besonders viele Beanstandungen aufgetreten sind, die stichwortartig dokumentiert wurden. Andere Kategorien der heuristischen Evaluation waren „Recognition Rather Than Recall”, „Flexibility and Minimalist Design”, „Aesthetic and Minimalist Design” und „Pleasurable and Respectful Interaction with the User”.

7.3.1 Visibility of System Status

Anzahl der Beanstandungen

Kategorie	Experte A	Experte B	Experte C
Catastrophe	1	2	0
Major	1	3	4
Minor	5	4	3

Beanstandungen

- Bei Änderungen des Granularitätslevels kommt es mitunter zum Systemstillstand bis zu 15 Sekunden bis der Bildschirminhalt aktualisiert wird. In diesem Zeitraum erhält der User kein Feedback über den Systemzustand. (2x catastrophe, 1x major)
- Die Oberfläche verhält sich in vielen Fällen noch nicht erwartungskonform, da Updates des Bildschirminhalts nach Benutzung von Slidern oder Buttons in einigen Fällen nicht oder nur teilweise durchgeführt werden (1x catastrophe, 1x major)
- Response Zeiten im allgemeinen zu hoch. Darstellung wirkt träge. (3x major)
- Keine Tooltips für Buttons oder ähnliche Elemente beim Mouse-Over (1x major)
- Der Selection-Status eines Dokuments wird mithilfe eines grün/blauen kreisförmigen Buttons dargestellt. (1x major)

Erläuterung: Die runde Form ruft eine Assoziation zum Radio Button hervor. Dies ist irreführend da keine exklusive Selektion vorliegt, sondern multiple selection.

- Weiterhin ist aus grün/blau nicht intuitiv ersichtlich, ob selektiert ist oder nicht (Was bedeutet grün? Was bedeutet blau?). Die Verwendung einer Checkbox oder eines 3D Buttons wäre sinnvoller.

- Bei höherer Granularität (gerade bei Level 6) erscheint der Button „verloren“ innerhalb der vertikal stark ausgedehnten Selection-Spalte des Dokuments. Hier könnte ein Button über die gesamte Zeilenhöhe mit starkem 3D Effekt oder Invertierung den Selektionsstatus besser darstellen und wählbar machen.

- Bei Selektion von Dokumenten in der Tabelle wird der entsprechende Punkt im Scatterplot nicht markiert, wenn er “unter” anderen Dokumenten liegt. Weiterhin ist schwer nachvollziehbar, welcher Punkt soeben durch die Auswahl in der Tabelle zusätzlich im Plot markiert wurde. Dasselbe gilt für die umgekehrte Variante, in der ein Dokument über einen Punkt im Scatterplot markiert wird. (1x major)

- Die verwendete Terminologie deckt sich nicht mit der Anwendungsdomäne. „Dimensions“ (2x minor), „Global/Local“ (1x minor), „Bean“ (1x minor), „Sphere“ (1x minor)

- Sprachliche Erklärung in der Legende des Scatterplots durch “Bean” etc ist für den User nicht leicht nachvollziehbar (1x minor)

- Kein Feedback vonseiten des Mauscursors über welcher Art von Element oder Auswahlmöglichkeit er sich befindet. (3x minor)

- Veränderungen an den Parametern des Scatterplots werden erst nach Drücken des Apply-Buttons in der grafischen Darstellung umgesetzt, da ein direktes Neuzeichnen wahrscheinlich zu zeitintensiv ist. Als Feedback für den User sollte daher nach einer Parameteränderung im Scatterplot angezeigt werden, dass die grafische Darstellung nicht mehr den aktuellen Einstellungen entspricht. (1x minor)

- Nicht jeder Screen beginnt mit einem Titel oder Seitenkopf, der den Bildschirminhalt beschreibt. (1x minor)

- Bezeichnungen und Kommandos werden nicht innerhalb des ganzen Systems einheitlich und durchgängig verwendet (1x minor)

7.3.2 Consistency and Standards

Anzahl der Beanstandungen

Kategorie	Experte A	Experte B	Experte C
Catastrophe	1	4	0
Major	1	2	1
Minor	5	5	3

Beanstandungen

- Farbe der Beschriftung der Balken auf Level 4 ist immer schwarz, auch bei dunkelblauem Hintergrund (schlechte Lesbarkeit). (1x catastrophe, 1x major)
- Auf Level 6 wird die Spalte „Visualization“ weiterhin angezeigt, obwohl sie auf Level 6 keine Funktion hat. Es kommt zu Inkonsistenzen der Darstellung bei dem Versuch die Spalte zu bewegen. (1x catastrophe)
- Die Visualisierung der Dokumente auf Level 5 durch farbige Kästchen ist oftmals nicht anklickbar. Im Falle eines erfolgreichen Klicks funktioniert oftmals das erwartete Texthighlighting nicht. Es findet keine Neupositionierung des dargestellten Ausschnitts entsprechend der Selektion in der Text-Spalte statt. (1x catastrophe)
- Es werden Blautöne für Textdarstellung verwendet. (2x catastrophe)
- Farben, Schriften, Formen entsprechen nicht den gängigen Industrie- und Unternehmens-Standards. (1x major)
- Im Magic Lense Configuration Dialog Fenster: Button für negative Antwort („Abort“) steht links und positive Antwort („Get this lense“) rechts. Nicht erwartungskonform.(1x major)
- Die ausgewählte Sortierreihenfolge in der Visualization-Spalte wird nicht in der Spaltenbeschriftung dargestellt. Üblich ist hier normalerweise ein Pfeil/Dreieck das nach oben/unten üblich ist. (1x major)

- Es ist kein Resizing der Breite und kein horizontales Scrolling der Spalten in der Granularity Table möglich. Dies wäre zumindest bei der Text-Spalte sinnvoll. (1x minor)
- Magic Lense Configuration Dialog Fenster: Es werden zu viele unterschiedliche Schriftgrößen verwendet. (1x minor)
- Magic Lense Configuration Dialog Fenster: Es gibt keinen Fenstertitel. (1x minor)
- Magic Lense Configuration Dialog Fenster: Integer sind nicht rechtsbündig ausgerichtet.
  (2x minor)
- Nicht erwartungskonforme Tastaturbelegung: Cursor hoch/runter verändert Granularitätslevel. Besser: Cursor hoch/runter für Dokumentenauswahl, Cursor links/rechts für lokales Granularitätslevel. (1x minor)
- Nicht jedes Fenster hat einen Titel (1x minor)
- Es gibt kein auffallendes visuelles Merkmal, das das aktive Fenster identifiziert (1x minor)
- Es werden die Farbtöne violett und rot verwendet, die nicht ausreichend weit voneinander im Farbspektrum entfernt sind (1x minor)
- Feldinhalte und Feldbeschriftung sind typographisch nicht voneinander getrennt (1x minor)
- Lange Zeichenketten oder Zahlen werden nicht in einzelne Blöcke aufgeteilt (?) (1x minor)
- Darstellung von Zahlen im ScatterPlot in manchen Fällen nicht lesbar, weil zu eng (1x minor)
- Im ScatterPlot liegen die Punkte in der Horizontalen manchmal vor dem Nullpunkt (1x minor)

8 Literaturverzeichnis

8.1 Veröffentlichungen zu GIS und GIS-Infrastruktur

[1:GeoBroker]: Web-Site „GeoBroker Brandenburg“ beim Fraunhofer ISST, http://www.isst.fhg.de

[1:InGeoForum]: Web-Site „InGeoForum“, http://www.ingeoforum.de

[1:MICUS]: Fornefeld M., Oefinger P.: Produktkonzept zur Öffnung des Geodatenmarktes, MICUS Management Consulting GmbH, Land NRW, September 2002, http://www.newmedianrw.de/downloads/Geodatenmarkt_MICUS_NRW_2002.pdf

8.2 Veröffentlichungen zu INVISIP/INSYDER

[2:Eibl et al.]: Eibl M., Klein P., Limbach T., Müller F., Reiterer H.: Visualization of Metadata Using the SuperTable+Scatterplot, ISI2002, Regensburg, 8.-10.Oktober 2002

[2:Göbel et al. 1]: Göbel S., Haist J., Reiterer H., Müller F.: INVISIP: Metadata-based Information Visualization Techniques to Access Geodata Archives and to Support the Site Planning Process , 3. CODATA Euro-American Workshop, 2002, Juli 10-11, Paris, France

[2:Göbel et al. 2]: Göbel S., Klein P.: Ranking Mechanisms in Meta-data Information Systems for Geospatial Data Conference for Developers of Geospatial Data Services over the Web, EOGEO 2002, 13.-15.Mai, Ispra, Italien

[2:INSYDER Web]: Web-Site „INSYDER“, http://www.insyder.com

[2:INVISIP]: Web-Site “INVISIP”: http://www.invisip.de

[2:Klein et al.]: Klein P., Müller F., Reiterer H., Eibl M.: Visual Information Retrieval with the SuperTable + Scatterplot, in: Proceedings of the 6th International Conference on Information Visualsation (IV 02), IEEE Computer Society, 2002, S.70-75.

[2:Mußler et al.]: Mußler G., Reiterer H., Mann T. M.: INSYDER - Information Retrieval Aspects of a Business Intelligence System, in: Knorz G., Kuhlen R. (Hg.): Informationskompetenz - Basiskompetenz in der Informationsgesellschaft, Proceedings des 7. Internationalen Symposiums für Informationswissenschaft, UKV Universitätsverlag Konstanz, Konstanz, 2000, S.127-143

8.3 Veröffentlichungen zu visuellen Suchsystemen und zum Document Retrieval

[3:Card et al.]: Card S.K., Mackinlay J.D., Shneiderman B.: Readings in Information Visualization. Using Vision to Think. Morgan Kaufmann Publishers, Inc, San Francisco, CA, 1999. 2002-11-21

[3:Fishkin, Stone]: Fishkin K., Stone M.: Enhanced dynamic queries via movable filters. In Human Factors in Computing Systems (CHI '95 Proceedings), pp. 415-420. New York: ACM Press, 1995

[3:Marchionini]: Marchionini G.: Information Seeking in Electronic Environments. Cambridge, UK: Cambridge University Press, 1995.

8.4 Veröffentlichungen zur heuristischen Evaluation

[4:Nielsen et al.]: Nielsen J., Mack R.: Usability Inspection Methods, John Wiley & Sons, 1994

[4:Weiss]: Weiss E.: Making Computers People-Literate, Jossey-Bass, 1994

[4:Xerox]: Pierotti D.: Heuristic Evaluation – A System Checklist, Xerox Corporation, 1995, http://www.stcsig.org/usability/topics/articles/he-checklist.html

8.5 Veröffentlichungen zu Usability-Questionnaires und Webbefragung

[5:Perlman]: Perlman G.: Web-Based User Interface Evaluation with Questionnaires, http://www.acm.org/~perlman/question.html, ACM, 1998

[5:Bosjnak, Tuten]: Bosjnak M., Tuten T. L.: Classifying Response Behaviors in Web-based Surveys, Journal of Computer-Mediated Communication, Vol 6, Issue 3, 2001

[5:Dillman et al.]: Dillman D. A.; Tortora R. D.; Conradt J.; Bowker D.: Influence Of Plain Vs. Fancy Design On Response Rates For Web Surveys, Joint Statistical Meetings, Dallas, 1998.

[5:WAMMI]: Web-Site “WAMMI”, http://www.nomos.se/wammi

8.6 Veröffentlichungen zu Methoden & Techniken der Usability-Evaluation

[6:Cugini, Laskowski]: Cugini J., Laskowski S.: Design of a File Format for Logging Website Interaction, NIST Special Publication 500-248, National Institute of Standards and Technology, April 2001

[6:Davis]: Davis, F. D.: Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology, MIS Quarterly (13:3), September 1989, pp.319-339.

[6:Gonzalez, Alvarez]: González Rodríguez M., Álvarez Gutierrez D.: Data Gathering Agents for Remote Navigability Testing, Proceedings of the SCI'2000 Conference (Systemics, Cybernetics and Informatics), Orlando, USA. 23th to 26th July 2000.

[6:Hartson, Castillo]: Hartson H. R., Castillo J.: Remote Evaluation for Post-Deployment Usability Improvement. Proceedings of the Working Conference on Advanced Visual Interface (AVI'98)

[6:Hassenzahl et al.]: Hassenzahl M., Platz A., Burmester M., Lehner K.: Hedonic and Ergonomic Quality Aspects Determine a Software's Appeal. In Proceedings of CHI 2000, ACM Press, 2000

[6:Hilbert et al. 1998]: Hilbert D.M., Redmiles D.F.: Agents for Collecting Application Usage data Over the Internet, Proceedings of the Second International Conference on Autonomous Agents, Minneapolis/St. Paul, MN, ACM, May 10-13, 1998.

[6:Hilbert et al. 1999]: Hilbert D.M., Redmiles D.F.: Extracting Usability Information from User Interface Events, Technical Report UCI-ICS-99-40, Department of Information and Computer Science, University of California, Irvine.

[6:Hong et al.]: Hong J., Landay J.: WebQuilt: A Framework for Capturing and Visualizing the Web Experience, Proceeding of WWW 10, Hong Kong, May 2001

[6:Ludi]: Ludi S.: Macromedia Director as a Prototyping and Usability Testing Tool, ACM Crossroads Xrds 6-5, http://www.acm.org/crossroads/xrds6-5/macromedia.html

[6:Nielsen 1]: Nielsen J.: Usability-Engineering, Academic Press, Boston MA, USA, 1993.

[6:Nielsen 2]: Nielsen J.: A Mathematical Model of the Finding of Usability Problems. Proceedings of the INTERCHI'93: Human Factors in Computing Systems, Amsterdam, The Netherlands, April 1993, pp. 206-213.

[6:Nielsen 3]: Nielsen J.: Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier. Web-Site http://www.useit.com, Nielsen Norman Group, 1994

[6:OCLC]: OCLC Web Site, http://www.oclc.org

[6:Schnell et al.]: Schnell R., Hill P., Esser E.: Methoden der empirischen Sozialforschung, 6. Aufl., Oldenbourg, München, 1999

[6:Schulz et al.]: Schulz E., Van Alphen M., Rasnake W.: "Discovering User-Generated Metaphors Through Usability Testing". Proceedings of the Second International Conference on Cognitive Technology. Aizu, Japan 1997.

Erklärung

Ich versichere hiermit, dass ich die anliegende Arbeit mit dem Thema „Usability-Evaluation im Rahmen von INVISIP“ selbständig verfasst und keine anderen Hilfsmittel als die angegebenen benutzt habe. Die Stellen, die anderen Werken dem Wortlaut oder dem Sinne nach entnommen sind, habe ich in jedem einzelnen Falle durch Angabe der Quelle, auch der benutzten Sekundärliteratur, als Entlehnung kenntlich gemacht.

Diese Arbeit wird nach Abschluss des Prüfungsverfahrens der Universitätsbibliothek Konstanz übergeben und ist durch Einsicht und Ausleihe somit der Öffentlichkeit zugänglich. Als Urheber der anliegenden Arbeit stimme ich diesem Verfahren zu / ~~nicht zu~~ *).

Konstanz, ... Unterschrift .....

*) Nichtzutreffendes bitte streichen.