

Fokus-und-Kontext-Techniken zur intelligenten Vergrößerung von graphischen Benutzungsoberflächen

Christiane Taras und Thomas Ertl

Institut für Visualisierung und interaktive Systeme, Universität Stuttgart, {taras, ertl}@vis.uni-stuttgart.de

Zusammenfassung

Der PC wird immer mehr zu einem unverzichtbaren Mittel für die tägliche Arbeit und die Kommunikation mit anderen Menschen. Gleichzeitig wächst durch die zunehmende Alterung unserer Gesellschaft die Zahl der leicht bis mittelstark sehbehinderten Menschen. Für diese ist es häufig mühsam, am PC mit den vorgegebenen Darstellungsgrößen effizient zu arbeiten. Die Vergrößerung der Bildschirmausgabe kann hier Abhilfe schaffen. Durch eine Untersuchung der zurzeit etablierten Vergrößerungstechniken, zeigte sich allerdings, dass bisher noch keine vollkommen zufriedenstellende Technik am Markt existiert. Aus diesem Grund haben wir ein neues Konzept für die Vergrößerung graphischer Benutzungsoberflächen entwickelt, welches wir in diesem Beitrag vorstellen. Das neuentwickelte Konzept orientiert sich an Fokus-und-Kontext-Techniken und legt besonderen Wert darauf, dass jeweils die Elemente der Benutzungsoberfläche vergrößert werden, die für den Benutzer in der aktuellen Arbeitssituation interessant sind. Das neue Konzept verzichtet auf eine permanente Vergrößerung aller Elemente, um alle durch die Anwendungen präsentierten Informationen auf dem Bildschirm darstellen zu können. Bei der Vergrößerung der für den Benutzer interessanten Elemente wird auf semantische Zusammenhänge geachtet, wie sie z.B. zwischen Einträgen eines Menüs existieren. Die ersten Erfahrungen mit einer Umsetzung des neuen Vergrößerungskonzeptes zeigen, dass dieses deutliche Vorteile gegenüber den etablierten Techniken hat.

1 Einleitung

Nicht zuletzt durch die stetig wachsende Auflösung von Standard-Monitoren ist die vergrößerte Darstellung von graphischen Benutzungsoberflächen seit einigen Jahren ein Thema in der Softwareentwicklung. Allerdings wurde dieser Bereich bisher als Randbereich zur Unterstützung der stark sehbehinderten PC-Benutzer betrachtet. Durch die zunehmende Alterung unserer Gesellschaft und die wachsende Rolle des PC als Arbeits-, Kommunikations- und Informationsmittel in allen Altersgruppen, wird die Softwareentwicklung zukünftig mit einer wachsenden Gruppe von älteren und dadurch leicht bis mittelstark sehbehinderten PC-Benutzern konfrontiert werden. Bei diesen Benutzern steigt das Bedürfnis nach Vergrößerungstechniken, die die gewohnte Arbeitseffizienz erhalten. In den letzten Jahren haben sich verschiedene Vergrößerungstechniken etabliert, die allerdings alle einige Nachteile aufweisen. Aus unseren Erfahrungen mit den etablierten Techniken und einer strukturierten Untersuchung dieser Techniken, haben wir ein neues Konzept zur Vergrößerung von graphischen Benutzungsoberflächen entwickelt, das an bekannte Fokus-und-Kontext-Techniken angelehnt sind. Zur Untersuchung des entwickelten Konzepts haben wir dies prototypisch anhand des Webbrowsers Mozilla Firefox umgesetzt.

In den folgenden Kapiteln präsentieren wir unsere Ergebnisse zur Untersuchung der etablierten Techniken, stellen das von uns entwickelte Konzept und dessen prototypische Implementierung vor und diskutieren die ersten Erfahrungen bei der Anwendung des neuen Konzeptes.

2 Diskussion aktueller Vergrößerungstechniken

In den letzten Jahren haben sich verschiedene Vergrößerungstechniken etabliert, die man unter den folgenden Punkten zusammenfassen kann:

1. Erhöhung der Standard-Schriftgröße
2. Vergrößerung eines rechteckigen Bildschirmausschnittes (Bildschirmlupe)

3. Anwendungsbezogene Zoomfunktion (z.B. in Webbrowsern und Büro-Anwendungen)
4. Vergrößerung des gesamten Bildschirminhaltes auf ein Vielfaches der Bildschirmgröße
5. Verringerung der Bildschirmauflösung

Diese Techniken wurden nach den folgenden, für Benutzer besonders wichtigen, Kriterien untersucht:

1. Vergrößerung auf eine bestimmte Schriftgröße

Bei der Arbeit am PC mit eingeschränkter Sehfähigkeit ist es wesentlich, dass Text in großer, gut erkennbarer Schrift dargestellt ist. Die zum größten Teil etablierte Vergrößerung um einen Faktor ist eigentlich nicht im Sinne des Benutzers, da dieser auf eine bestimmte Mindestgröße der Schrift angewiesen ist. Das Ergebnis der Anwendung eines Vergrößerungsfaktors ist allerdings von der Ausgangsschriftgröße abhängig und kann somit sehr unterschiedlich ausfallen. Die Vergrößerung auf eine bestimmte Schriftgröße kann zwar dazu führen, dass z.B. normaler Text genauso groß dargestellt wird, wie Überschriften. Da für solche Unterscheidungen aber meist noch andere Gestaltungsmittel eingesetzt werden (z.B. fette Schrift) ist dies kein wesentlicher Nachteil. Zur Unterstützung verschieden hoher Einschränkungen der Sehfähigkeit muss die Schriftgröße möglichst variabel von sehr geringen bis zu sehr hohen Vergrößerungen einstellbar sein.

2. Gute Qualität des vergrößerten Textes

Der vergrößerte Text muss möglichst klar und nicht verschwommen oder verpixelt dargestellt sein, um möglichst gut erkennbar zu sein.

3. Vergrößerter Mauszeiger

Da der Mauszeiger das am häufigsten verwendete Mittel ist, um Elemente auf der graphischen Benutzungsoberfläche auszuwählen, muss er für den Benutzer auf den ersten Blick gut sichtbar sein. Somit muss er immer in einer für den Benutzer ausreichenden Größe dargestellt werden.

4. Vergrößerung der Benutzungsoberfläche (4a) und des Anwendungsdokumentes¹ (4b)

Zur effizienten Arbeit am PC muss eine Vergrößerung sowohl für die Benutzungsoberfläche als auch für die Anwendungsdokumente zur Verfügung stehen.

5. Gute Übersicht über den gesamten Bildschirminhalt / Sichtbarkeit aller Informationen

Eine Vergrößerung der auf dem Bildschirm dargestellten Informationen darf nicht dazu führen, dass Informationen, wie z.B. Sicherheitshinweise oder Abfragedialoge (die häufig auch andere Anwendungen blockieren), außerhalb des Sichtbereichs des Benutzers liegen.

6. Wenig horizontales Scrollen beim fortlaufenden Lesen von Dokumenten

Das fortlaufende Lesen von Dokumenten sollte für den Benutzer möglichst einfach sein, um eine möglichst hohe Arbeitseffizienz zu erreichen. Dabei sollte durch eine Vergrößerung möglichst keine Notwendigkeit zum horizontalen Scrollen entstehen, da dies für den Benutzer sehr ermüdend und häufig auch verwirrend ist. (Siehe dazu auch Beckmann & Legge 1996).

7. Viel Platz für das Anwendungsdokument

Der wesentliche Grund für die Arbeit am PC ist das Lesen und Bearbeiten der Anwendungsdokumente. Deshalb muss zur Darstellung dieser Dokumente möglichst viel Platz zur Verfügung stehen.

¹ Anwendungsdokument: Das Dokument, das innerhalb der Benutzungsoberfläche einer Anwendung dargestellt wird z.B. ein Word-Dokument, eine Webseite, ein Textdokument.

8. Schnelle Verfügbarkeit, leichte Änderbarkeit der Vergrößerungsstufe

Da die Verschlechterung der Sehfähigkeit in den meisten Fällen schleichend voranschreitet und dem Benutzer häufig gar nicht bewusst ist, sollte es dem Benutzer so einfach wie möglich gemacht werden, am PC mal eine größere Schriftgröße bzw. eine höhere Vergrößerung auszuprobieren.

9. Vergrößerung von Icons

Icons sollen die Verständlichkeit einer Benutzungsoberfläche erhöhen und die Arbeit mit einer Oberfläche beschleunigen. Dies kann natürlich nur gelingen, wenn sie vom Benutzer auch gut erkannt werden können. Deshalb müssen auch Icons in einer möglichst guten Qualität auf die vom Benutzer bevorzugte Größe skaliert werden können.

10. Gemeinsame Vergrößerung logisch zusammenhängender Informationen

Für eine effiziente Arbeitsweise ist es wichtig, dass Informationen, die für den Benutzer logisch zusammengehören (wie z.B. alle Einträge eines Menüs oder eine Schaltfläche und deren Tooltip), auch gemeinsam vergrößert werden.

11. Erhaltung des Layouts

Die Erhaltung des ursprünglichen Layouts ist aus mehreren Gründen wichtig. Ausgehend davon, dass sich die Sehfähigkeit des Benutzers nach und nach verschlechtert und der Benutzer die Arbeit mit einer Anwendung seit längerer Zeit gewohnt ist, sollte sich die Anordnung der Elemente der Oberfläche auch bei Vergrößerung nicht wesentlich verändern, um keine zusätzliche Belastung zu erzeugen. Aber auch wenn der Benutzer die Anwendung erst kennenlernt, kann eine Änderung des Layouts der Benutzungsoberfläche hinderlich sein, da Programmhilfen die Position von Menüs oder Steuerelementen häufig absolut angeben (wie z.B. „oben auf der linken Seite“) oder auf Screenshots der Originaldarstellung anzeigen. Bei Dokumenten ist die Erhaltung des Layouts sinnvoll um dem Benutzer die Möglichkeit zu geben, gut strukturierte Dokumente erstellen zu können und da durch das Layout häufig auch logische Zusammengehörigkeiten dargestellt werden

Die folgende Tabelle zeigt die Untersuchungsergebnisse in einer Übersicht.

Technik	Kriterium											
	1	2	3	4a	4b	5	6	7	8	9	10	11
1	ja	gut	nein	teilweise	möglich	ja	nein	gering	nein	möglich	ja	nein
2	nein	eher schlecht	nein ²	ja	ja	ja	nein	normal	ja	ja	nein	ja
3 ³	nein	gut	nein	nein	ja	teilweise	teilweise	normal	ja	teilweise	ja	teilweise
4 ³	nein	eher schlecht	ja	ja	ja	nein	nein	normal	ja	ja	ja	ja
5	nein	schlecht	ja	ja	ja	ja	nein	gering	eher nicht	ja	ja	teilweise

Tabelle 1 Übersicht der Ergebnisse zur Untersuchung der etablierten Techniken

² Der Mauszeiger wird zwar innerhalb der Bildschirmlupe vergrößert, dies nutzt dem Benutzer bei der Auswahl von Elementen aber nur wenig.

³ Für Kriterium 3 wurden die Anwendungen Internet Explorer, Mozilla Firefox und Microsoft Word genauer untersucht und für Kriterium 4 die Anwendungen ZoomText Magnifier (www.zoomtext.de) und NLarge (www.codeplex.com/NLarge)

Die Untersuchung zeigte, dass keine der etablierten Techniken alle genannten Kriterien erfüllt. Beispielsweise ist das fortlaufende Lesen von Dokumenten mit den Techniken 2, 4 und 5 mit hohem Scrollaufwand verbunden, 1 und 5 führen meist zu einem hohen Informationsverlust, da durch die permanente Vergrößerung aller Elemente auf der Größe des normalen Bildschirms nur wenig Platz für das Anwendungsdokument bleibt. Zudem ist die Anwendung von Technik 1 für den Benutzer meist sehr aufwändig. Technik 3 beschränkt sich meist auf die Vergrößerung des Anwendungsdokumentes und vergrößert die Benutzungsoberfläche selbst nicht. Die von Sehbehinderten wahrscheinlich am häufigsten genutzte Technik 4 hat das große Problem, dass Dialoge, die andere Anwendungen blockieren vom Benutzer häufig nicht gesehen werden können, da sie sich nicht im gerade auf dem Bildschirm angezeigten Bereich befinden. Da die Techniken von den Anwendungen bzw. vom Betriebssystem alle unabhängig voneinander behandelt werden ist auch keine sinnvolle Kombination möglich.

Um in Zukunft eine angemessene Unterstützung für die Arbeit am PC mit eingeschränkter Sehfähigkeit zu gewähren, muss als eine neue Technik entwickelt werden, die alle oben genannten Kriterien erfüllt und somit die Vorteile der etablierten Techniken vereint, deren Nachteile abbaut und vor allem schnell und überall gleichermaßen zur Verfügung steht.

3 Vorstellung des neuen Vergrößerungskonzepts

Das von uns neu entwickelte Vergrößerungskonzept soll alle, der in Kapitel 2 genannten Kriterien erfüllen. Es ist an bekannte Fokus-und-Kontext-Techniken angelehnt und baut im Wesentlichen auf folgenden drei Grundsätzen auf:

1. Weitestgehende Erhaltung der Original-Ansicht zur Erhaltung des Überblicks und Verbesserung der Orientierung (Kontext)
2. Vergrößerung der für den Benutzer gerade interessanten Bereiche mit Beachtung semantischer Zusammenhänge (Fokus)
3. Effektive Platznutzung

Ausgangspunkt der Vergrößerung soll jeweils die normale, also unvergrößerte Ansicht der Anwendungen sein. Die Übersicht soll dadurch gewährleistet werden, dass keine bzw. nur wenige Elemente permanent vergrößert werden. Dies erhält auch den Platz, der für das Anwendungsdokument zur Verfügung steht. Vergrößert muss nur werden, was den Benutzer in der aktuellen Arbeitssituation interessiert. Dies kann z.B. durch die Position des Mauszeigers oder des Eingabecursors, durch Eye-Tracking oder über eine Auswahl von Elementen durch den Benutzer ermittelt werden. Für bestimmte Elemente wie z.B. Titelleisten und Statusleisten sollte eine permanente Vergrößerung einstellbar sein.

Im Gegensatz zu bekannten Fokus-und-Kontext-Techniken soll nicht nur das gerade direkt fokussierte Element vergrößert werden, sondern auch Elemente, die logisch mit diesem Element eng zusammenhängen. Aktiviert der Benutzer beispielsweise ein Eingabefeld, so soll dieses inklusive der zugehörigen Schaltflächen und des Eingabecursors vergrößert werden. Bei Fokussierung eines Menüs soll nicht nur ein einzelner Menü-Eintrag vergrößert werden, sondern alle Einträge auf gleicher Ebene und auch die übergeordneten Einträge (siehe z.B. Abbildung 1). So kann der Benutzer schnell erkennen, wo er sich gerade befindet und welche anderen Auswahlmöglichkeiten ihm zur Verfügung stehen. Ein anderes Beispiel ist die Behandlung kleinerer Dialoge, wie z.B. Passwortabfragen, die meist mittig auf dem Bildschirm erscheinen. Da diese Dialoge nur wenig Platz einnehmen, alle ihre Inhalte semantisch stark zusammenhängen und für den Benutzer auch im Moment des Erscheinens des Dialoges interessant sind, sollen Dialoge sofort bei Erscheinen im Ganzen vergrößert werden (siehe Abbildung 4). Tooltips und Icons, die zu einem vergrößerten Element gehören müssen passend zum Element vergrößert werden.

Für die Umsetzung des neuen Konzeptes wurden zwei Vergrößerungsmöglichkeiten identifiziert. Die direkte Vergrößerung von Elementen in der Originalansicht und die indirekte Vergrößerung von Elementen mit Hilfe von Lupen. Abbildung 1 veranschaulicht beide Möglichkeiten.

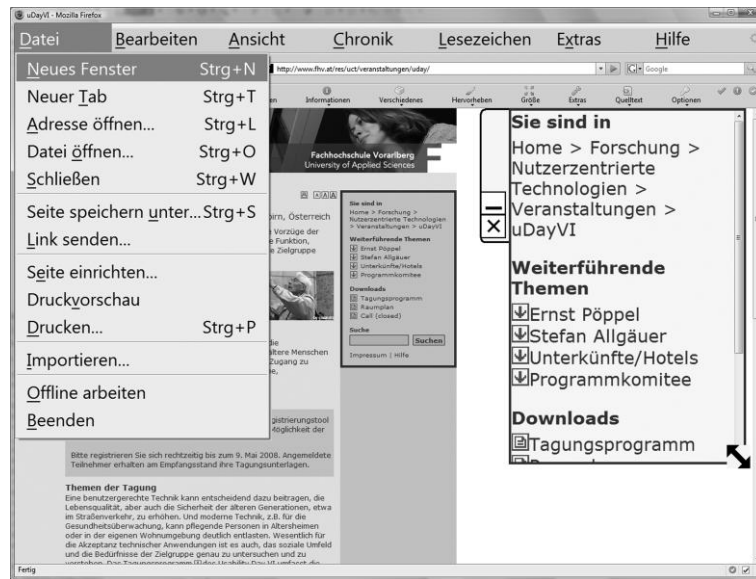


Abbildung 1 Darstellung einer Lupe (rechts) in Kombination mit der vergrößerten Menüleiste und einem vergrößerten Menü (der in der Lupe vergrößerte Bereich ist in der Webseite durch einen Rahmen markiert)

Die direkte Vergrößerung bedeutet eine Änderung der Größe der originalen Elemente der Oberfläche oder des Anwendungsdokumentes, was das Layout in gewissem Maße beeinflusst, aber keine zusätzlichen Bedienelemente erfordert. Die indirekte Vergrößerung basiert auf der Verdopplung der Elemente. Sie werden in der Originalansicht unvergrößert dargestellt und nur in einer Lupe vergrößert. Dies erhält zwar das Originallayout. Da die Lupe aber ein zusätzliches Bedienelement ist, werden hier möglicherweise Informationen überdeckt, die in der normalen Darstellung sichtbar wären. Durch die Größe und Auflösung aktueller Bildschirme steht aber meist genügend Platz für die Positionierung von Lupen zur Verfügung, was in Abbildung 1 deutlich zu erkennen ist. Der wesentliche Nachteil von Lupen ist, dass der Benutzer gezwungen wird, seine Fokussierung oft zwischen der Originalansicht und der vergrößerten Darstellung zu wechseln.

Für die direkte Vergrößerung ist eine Unterscheidung zwischen Elementen, die in der Benutzungsoberfläche direkt sichtbar sind und Elementen, die erst sichtbar werden notwendig. Elemente, die direkt sichtbar sind, müssen für den Benutzer ohne Probleme anvisierbar sein und dürfen so ihre Position bei Fokussierung, was ja zur Vergrößerung führt, nicht verändern. Bereits sichtbare Elemente müssen somit auch bei Vergrößerung auf ihre ursprüngliche Ausdehnung beschränkt werden. Elemente, die erst sichtbar werden und bei Verlust des Fokus wieder verschwinden, können schon kurz vor ihrem Erscheinen vergrößert werden. Somit führt eine Veränderung der absoluten Position dieser Elemente durch die Vergrößerung nicht zu einer Irritation beim Benutzer.

Sowohl bei der direkten also auch bei der indirekten Vergrößerung muss gewährleistet sein, dass der Benutzer in dem vergrößerten Bereich normal arbeiten kann. Die vergrößerten Elemente müssen die gleichen Funktionalitäten aufweisen, wie die originalen, unvergrößerten Elemente. So muss z.B. die Verwendung von Eingabefeldern und Schaltflächen oder auch das Markieren von Text genauso wie im unvergrößerten Zustand möglich sein.

4 Umsetzung des neuen Vergrößerungskonzepts

Die erste (prototypische) Umsetzung des neuen Vergrößerungskonzeptes basiert auf dem Webbrowser Mozilla Firefox und soll vor allem zur Evaluierung des Konzeptes dienen. Die Anwendung Mozilla Firefox bot sich für die erste Umsetzung an, da sie sehr leicht erweiterbar ist und ihre Darstellung durch die moderne XML-basierte Oberfläche auch sehr leicht beeinflussbar ist.

Die Änderung der Darstellung erfolgt größten Teils durch Beeinflussung von CSS (Cascading Style Sheet) Anweisungen. Teilweise muss aber auch in den DOM-Baum (Document Object Model) der Anwendung eingegriffen werden. Grundlage der Vergrößerung ist jeweils das Setzen des Schriftgröße-Attributs („font-size“). Elemente wie Icons, Schaltflächen und Eingabefelder werden durch Setzen der Größen-Attribute („width“ und „height“) passend skaliert bzw. in ihrer Ausdehnung begrenzt. In den hier gezeigten Abbildungen wurde die minimale Schriftgröße jeweils auf 30 px eingestellt. Da die standardmäßig verwendete Schriftgröße in der Benutzungsoberfläche meist 12 px beträgt, ist dies vergleichbar mit einem Vergrößerungsfaktor von 2,5.

In der aktuellen Umsetzung kann der Benutzer die minimale Schriftgröße über eine Tastenkombination verändern und die Vergrößerungsfunktion über eine Tastenkombination an- und abschalten. Beide in Kapitel 3 besprochenen Vergrößerungsmöglichkeiten wurden in der aktuellen Implementierung umgesetzt und können kombiniert werden, wie Abbildung 1 zeigt. Elemente, die bereits vor der Vergrößerung eine höhere Schriftgröße besitzen als die vom Benutzer vorgegebene Mindestschriftgröße, werden durch die Vergrößerung nicht beeinflusst.

Während der Implementierung zeigte sich, dass für Benutzungsoberflächen vor allem die direkte Vergrößerung sinnvoll ist, während sich bei Webseiten (aufgrund der Schwierigkeit semantische Zusammenhänge in Webseiten automatisch zu erkennen) vor allem die indirekte Vergrößerung eignet. Außerdem bieten Webseiten meist ausreichend Platz für eine Überlagerung durch Lupen ohne Informationen zu verdecken. Die direkte Vergrößerung wurde für Webseiten auch umgesetzt, führte aber aufgrund des schon genannten Problems zu häufigen, verwirrenden Wechseln zwischen vergrößertem und unvergrößertem Zustand verschiedener Bereiche. Aus den Erfahrungen mit anderen Anwendungen wie Eclipse und einfachen Editoren lässt sich allerdings schließen, dass sich für Dokumente häufig auch eine Vergrößerung im Ganzen eignet und ein Wechsel zwischen verschiedenen Ansichten für fortlaufendes Lesen und Layoutkontrolle sinnvoll ist.

4.1 Direkte Vergrößerung in der Benutzungsoberfläche

Elemente der Benutzungsoberfläche, die direkt sichtbar sind, werden vergrößert sobald der Benutzer eine kurze Zeit mit dem Mauszeiger über ihnen verweilt. Verlässt der Mauszeiger die Elemente wieder, werden diese in den Originalzustand zurückgesetzt. Hierfür werden die originalen Werte der veränderten Attribute zwischengespeichert. Aufpoppende Elemente, wie Menüs, Dialoge und Tooltips werden bei Erscheinen vergrößert. Bewegt sich der Benutzer durch Untermenüs, so bleiben auch die übergeordneten Menüs vergrößert. Per Tastendruck kann die Vergrößerung für bestimmte Elemente wie z.B. die Statusleiste festgesetzt und wieder gelöst werden.

Um ein gutes Zielen auf Elemente zu ermöglichen, wurde die Breite von Schaltflächen in den Werkzeugleisten begrenzt. Wie Abbildung 2 zeigt, kann dies natürlich dazu führen, dass die Beschriftung im vergrößerten Zustand nicht mehr auf die Schaltflächen passt. Deshalb mussten die Beschriftungen durch die Anwendung des Attributes „crop“ verkürzt werden. Um dem Benutzer dennoch die gesamte Beschriftung zugänglich zu machen, wurde diese in die Tooltips der Schaltflächen eingefügt.

Um trotz der Größenbeschränkung für direkt sichtbare Elemente auch im vergrößerten Zustand möglichst viel Informationen darstellen zu können, wurde schon die unvergrößerte Darstellung der Werkzeugleisten und der Menüleiste gegenüber der Originaldarstellung leicht verändert. Wie Abbildung 3 zeigt, wird für solche Elemente meist nicht die gesamte Breite des Anwendungsfensters ausgenutzt, um die nötigen Mausbewegungen zu beschränken. Würde man die Menüleiste aus der Originalansicht heraus vergrößern, wären nur einige Buchstaben der Menünamen sichtbar, obwohl die gesamte Breite der Menüleiste genügend Platz für eine unverkürzte Darstellung bieten würde. Durch die Änderung der Originaldarstellung, wie sie in Abbildung 3 zu sehen ist, wird der gesamte zur Verfügung stehende Platz ausgenutzt ohne den Benutzer durch eine Änderung der Position der Menünamen bei Vergrößerung zu irritieren. Da sich die Vorbereitung des Menüs nur auf die absolute Position der Menünamen auswirkt und nicht auf deren relative Positionierung zueinander, ergeben sich hier auch keine Nachteile. Die Streckung der Einträge auf die gesamte Breite des Anwendungsfensters wird auch bei den Werkzeugleisten angewendet. Außerdem wird hier zusätzlich von einer vertikalen Ausrichtung der Icons zur Beschriftung auf eine

horizontale Ausrichtung umgestellt, um noch mehr Platz für die vergrößerte Beschriftung zu schaffen (siehe Abbildung 2).



Abbildung 2 Vergrößerung von Schaltflächen auf einer Werkzeugleiste mit Tooltip

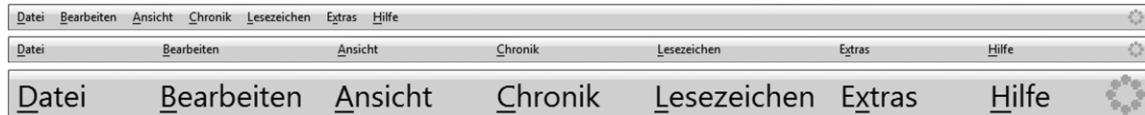


Abbildung 3 Vorbereitung und Vergrößerung der Menüleiste (oben: normale Darstellung der Menüleiste; Mitte: vorbereitete Menüleiste; unten: vergrößerte Menüleiste)

4.2 Indirekte Vergrößerung über Lupen in Webseiten

Für die indirekte Vergrößerung wurde die Benutzungsschnittstelle der Semantic Lens wieder aufgegriffen (Rotard et al. 2007). Eine neue Lupe kann über eine Tastenkombination oder per Doppelklick auf einen Abschnitt in einer Webseite geöffnet und dann frei innerhalb der Webseite positioniert werden. Wie Abbildung 1 zeigt, eignet sich meist der rechte Rand von Webseiten besonders gut für die Positionierung der Lupen, da dieser in aktuellen Webseiten häufig nicht genutzt wird.

Jede neue Lupe besitzt eine eigene Rahmenfarbe, über die der in der Lupe vergrößerte Bereich in der Webseite wiedergefunden werden kann. Verweilt der Benutzer eine kurze Zeit mit dem Mauszeiger über einem Bereich der Webseite, so wird dieser vergrößert in die gerade aktive Lupe geladen und mit der Rahmenfarbe der Lupe markiert. Da der gesamte Inhalt des Bereiches aus dem DOM-Baum der Webseite kopiert und in die Lupe geladen wird, können z.B. auch Eingabefelder oder Links innerhalb der Lupe verwendet werden und verhalten sich wie im originalen Abschnitt.

Der Benutzer kann für eine Webseite mehrere Lupen gleichzeitig öffnen, um verschiedenen Bereiche zu vergrößern. Die Benutzungsschnittstelle der Lupen ähnelt der von normalen Fenstern, somit kann jede Lupe minimiert, in ihrer Größe frei verändert und auch geschlossen werden. Beim Wechsel auf eine andere Seite können die Lupen automatisch gelöscht werden oder erhalten bleiben, sodass sie nicht neu geöffnet und positioniert werden müssen. Dieses Verhalten kann der Benutzer beeinflussen.

Die Lupen selbst werden als absolut positionierte DIVs direkt in die Webseiten eingebaut.

4.3 Vergrößerung von Dialogen

Um auf das Erscheinen eines Dialoges reagieren zu können, wurde für die aktuelle Firefox-Erweiterung die Schnittstelle „nsIWindowMediatorListener“ implementiert. Bei Erscheinen eines Dialoges wird zunächst dessen wahrscheinliche Größe nach Vergrößerung aller Dialogelemente ermittelt. Überschreitet diese eine gewisse Maximalgröße (Bildschirmgröße – Rand), so wird der Dialog normal angezeigt und später Aufgrund seiner Strukturierung in horizontale und vertikale Bereiche („hbox“, „vbox“) ähnlich wie in Kapitel 4.1 beschrieben vergrößert. Passt der vergrößerte Dialog in die vorgegebene Maximalgröße werden alle seine Elemente sofort vergrößert. Wie in Kapitel 3 beschrieben muss dabei nicht auf die vorherigen Ausdehnungen der Dialogelemente geachtet werden. Um eine optimale Platzausnutzung zu gewährleisten wird der Dialog danach auf die Größe seines Inhaltes zusammengeschrumpft, wodurch überflüssige Leerräume entfernt werden. Um Eingabe-Dialoge kompakt zu halten, wurde für Eingabefelder eine sinnvolle Größenbeschränkung eingeführt.



Abbildung 4 Vergrößerter Dialog zur Anmeldung an einem Webserver

5 Diskussion der Ergebnisse und zukünftige Arbeit

Die ersten Erfahrungen bei der Anwendung des neuen Vergrößerungskonzepts zeigen, dass dieses deutliche Vorteile gegenüber den etablierten Techniken hat. Die Übersicht über den Bildschirminhalt bleibt erhalten und dem Benutzer entgehen keine Informationen. Alle aufpoppenden Dialoge, Menüs und Tooltips sind sichtbar und werden auch vergrößert. Die Vergrößerung basiert nicht auf einem Vergrößerungsfaktor, sondern auf einer vom Benutzer vorgegebenen Mindestschriftgröße. Dabei werden auch Icons passend zur Änderung der Schriftgröße skaliert. Da die Vergrößerung der Oberflächenelemente nicht permanent besteht, bleibt für das Anwendungsdokument genauso viel Platz wie ohne eine Vergrößerungstechnik.

Bei der Umsetzung des neuen Konzeptes muss besonders darauf geachtet werden, dass durch die selektive Vergrößerung kein störendes Flackern durch einen ständigen Wechsel von vergrößertem und unvergrößertem Zustand eines Elementes erzeugt wird. Für Benutzungsoberflächen konnte dies in der aktuellen Umsetzung gut erreicht werden. Bei der Verwendung des Vergrößerungskonzeptes für Webseiten zeigten sich hier allerdings Schwierigkeiten. Um diesem entgegenzuwirken soll die aktuelle Umsetzung deshalb noch um eine zusätzliche Selektionsmöglichkeit für zu vergrößernde Elemente erweitert werden, die stärker vom Benutzer beeinflusst werden kann.

In den kommenden Monaten soll die aktuelle Umsetzung in einer Benutzerstudie evaluiert werden. Sofern das neue Konzept von den Benutzern gut angenommen wird, soll eine Umsetzung für die gesamte Windowsoberfläche erfolgen. Hierfür soll die MSAA-Schnittstelle und UI-Automation genutzt werden. Weiterhin wird die Kombination mit einem Bildschirmoverlay zur Unterstützung von Menschen mit Farbenblindheit untersucht werden.

6 Literaturverzeichnis

- Beckmann, P. J. & Legge, G. E. (1996). Psychophysics of Reading: XIV. The Page Navigation Problem in Using Magnifiers. In *Vision Research*, Volume 36, Issue 22, 3723-3733.
- Rotard, M., Giereth M. & Ertl, T. (2007). Semantic Lenses: Seamless Augmentation of Web Pages with Context Information from Implicit Queries. In *Computers and Graphics* 31(3), 361-369. Elsevier