

Unterstützung des Benutzungsoberflächen-Designs durch eine Kritikkomponente

Uwe Malinowski, Markus Stolze
Siemens AG, University of Colorado at Boulder

Zusammenfassung

Beim Entwurf grafischer Benutzungsoberflächen muß eine überwältigende Menge von Informationen berücksichtigt werden. Traditionell stehen diese auf Papier als Entwurfsrichtlinien oder Styleguides zur Verfügung und sind damit nur mühsam zugänglich. Eine Kritikkomponente kann durch aktive Präsentation der in einer bestimmten Designsituation benötigten Informationen den Designer unterstützen. Auf der Basis von Erfahrungen aus verschiedenen Projekten werden verschiedene Möglichkeiten der Informationspräsentation diskutiert.

1 Motivation

Der Entwurf einer grafischen Benutzungsoberfläche ist ein aufwendiger Prozeß. In den letzten Jahren sind verschiedene Werkzeuge entwickelt worden, die den Entwurf einer Benutzungsoberfläche vereinfachen. SX/Tools [7] und NextStep [5] sind Beispiele solcher Systeme, die dem Designer erlauben, die Benutzungsoberfläche per direkter Manipulation zu erstellen. Derartige Werkzeuge können zwar den Entwurfsprozeß beschleunigen, eine hohe Qualität der erstellten Benutzungsoberflächen ist damit aber noch nicht garantiert.

Im Unterschied zum Entwurf einer Kommando-orientierten Benutzungsoberfläche hat der Designer beim Entwurf einer grafischen Benutzungsoberfläche wesentlich mehr Gestaltungsmöglichkeiten. Wenn immer alle Freiheitsgrade ausgeschöpft würden, würde dies zu einer Überforderung der Endanwender führen. Um dieser Entwicklung entgegenzuwirken sind unterschiedliche nationale und firmeninterne Entwurfsrichtlinien und Styleguides entwickelt worden (z.B. SNI Grafik-Styleguide 2.0 von 1993). Die Nützlichkeit dieser Informationen ist jedoch dadurch beschränkt, daß sie normalerweise nur in der Form von Handbüchern zur Verfügung stehen, was bei der Menge und Vielfalt der Informationen, nicht angemessen ist.

In letzter Zeit sind verschiedene Ansätze verfolgt worden, die Entwurfsrichtlinien und Fallbeispiele "online" zur Verfügung zu stellen (z.B. [11], [1]). Aber auch die Möglichkeit mittels Datenbankabfragen und Hypertext-Browsing auf Entwurfsrichtlinien zugreifen zu können bringt noch relativ wenig Unterstützung für den Designer. Suchprozesse werden zwar vereinfacht, aber der Designer muß die

Initiative ergreifen, d.h. er muß wissen, daß ihm Informationen fehlen. Weiterhin muß er wissen, welche Informationen ihm fehlen, um danach suchen zu können. Eine weitere Schwäche dieses Ansatzes ist, daß bei der Formulierung der Anfrage nach Informationen wenig Unterstützung gegeben wird, so daß der Designer auch wissen muß, wie auf die Information zugegriffen werden kann.

In Unterschied zu den bisher erwähnten *passiven* Unterstützungstechniken gibt es Methoden, die eine *aktive* Rolle im Designprozeß übernehmen. Constraints beschränken aktiv die Freiheitsgrade in einer bestimmten Designsituation, während eine Kritikkomponente aktiv Informationen zur Verfügung stellt, die in einer bestimmten Situation von Bedeutung sind. Die aktive Rolle geht aber nicht so weit, daß versucht wird, das Design der Benutzungsoberfläche zu automatisieren, was z.B. bei UIDE [13] oder HUMANOID [10] das Ziel ist.

Constraints sind ein wichtiges Element von direkt-manipulativen Systemen. Die grundlegende Idee ist, daß Constraints nur „sinnvolle“ Operationen ermöglichen und damit dem Benutzer die Auswahl von Operationen erleichtert. Constraints werden jedoch dann problematisch wenn die Bestimmung der „sinnvollen“ Operationen nicht eindeutig ist. In diesen Fällen kann es passieren, daß Designer eine Operation ausführen möchten, welche das System nicht gestattet. Constraints sind deswegen hauptsächlich geeignet für die Kodierung von einfachen und „sicheren“ Regeln. Im Benutzungsoberflächen-Design sind mehrdeutige und zum Teil widersprüchliche Designregeln jedoch relativ häufig, so daß sich Constraints als Unterstützungstechnik weniger gut eignen.

Eine Kritikkomponente schränkt die Möglichkeiten von Entwicklern nicht ein, sondern analysiert den momentanen Zustand des Entwurfs, weist Entwickler auf potentielle Probleme und Entwurfsoptionen hin und ermöglichen den Zugriff auf Erklärungen und Hintergrundinformationen [3]. Aufgrund dieser Informationen können Entwickler entscheiden, ob die „Kritik“ in der aktuellen Situation gerechtfertigt ist und sie ihren Entwurf anpassen wollen, oder ob sie die Kritik ignorieren wollen, da sie von falschen oder unvollständigen Voraussetzungen ausgeht. Indem der Designer auf mögliche Probleme aufmerksam gemacht wird, aber keine Entscheidung über deren Behandlung getroffen wird bleibt die Kontrolle beim Designer [2]. Aufgrund dieser Eigenschaft sind Kritikregeln besonders geeignet für die Kodierung von komplexeren und „unsicheren“ Zusammenhängen.

Im folgenden werden wir verschiedene Möglichkeiten und Probleme der Informationspräsentation durch eine Kritikkomponente diskutieren. Wir wollen jedoch jetzt schon darauf hinweisen, daß im Normalfall eine Integration der Kritikkomponente mit anderen Techniken und Komponenten sinnvoll ist. So können

Constraints -wenn einsetzbar- verhindern, daß eine falsche Designentscheidung überhaupt getroffen wird, was in jedem Fall schneller ist, als eine spätere Korrektur. Für eine optimale Unterstützung sollte dem Designer zudem ein Argumentationswerkzeug und ist eine Katalogkomponente zur Verfügung stehen. Eine Integration dieser Komponenten in einem Designwerkzeug wird als Domain-Oriented Design Environment bezeichnet [3].

2 Eine Kritikkomponente im Benutzungsoberflächen-Design

Idealerweise sollte eine Kritikkomponente die im jeweiligen Designkontext benötigte Styleguide-Information in geeigneter Form zur Verfügung stellen. Grundvoraussetzung zur Erreichung dieses Ziel ist, daß die Styleguide-Informationen erfolgreich in Regeln umgesetzt werden können, eine Spezifikation für die Benutzungsoberfläche in einer für das System verarbeitbaren Form vorliegt, und die Designziele vom Designer definiert werden können. Aber selbst wenn dies gelingt, gibt es noch etliche offene Fragen: Zu welchem Zeitpunkt ist die Präsentation einer Information am effektivsten? Wie soll die Information präsentiert werden? Wo soll die Information präsentiert werden? In verschiedenen Projekten sind verschiedenen Lösungsansätze zu diesen Problemen erarbeitet worden. Diese werden wir in den nächsten Abschnitten vergleichen und in Beziehung zueinander setzen.

2.1 Zeitpunkt der Informationspräsentation

Der frühestmögliche Zeitpunkt der Präsentation von Informationen, die sich auf eine Designaktion beziehen, ist während der eigentlichen Aktion des Benutzers. Im Zusammenhang mit Kritikkomponenten im Bereich Benutzungsoberflächen-Design gibt es hierzu noch keine Erfahrungen. Bei Textverarbeitungsprogrammen und CAD-Systemen ist Feedback durch Veränderung des Cursor oder Signalisieren des Zustands, der erreicht würde, wenn die Aktion in dem aktuellen Moment ausgeführt/abgeschlossen würde, üblich. Ein Werkzeug, welches den Entwurf von Dialogboxen unterstützt, könnte zum Beispiel beim Plazieren eines Textfeldes durch die Veränderung des Cursors anzeigen, wenn dieses zu dicht neben einem anderen Objekt plaziert würde.

Die nächste Möglichkeit ist die Vermittlung einer Information unmittelbar nach der Aktion des Designers. Diese traditionelle Vorgehensweise trägt der Forderung Rechnung, daß die Information so früh wie möglich zur Verfügung gestellt werden soll, um nicht auf einem fehlerhaften Zustand aufzubauen, wenn man es vermeiden kann. In FRAMER [8], einem Werkzeug welches den Design von Symbolics Benutzungsoberflächen unterstützt, wird der Benutzer zum Beispiel direkt nach der

Verschiebung von Fensterbereichen auf nicht erlaubte Überlappungen der bewegten Bereiche hingewiesen.

Es ist offensichtlich, daß einige Analysen nicht unmittelbar nach einem Designschritt durchgeführt werden können, beziehungsweise in der Situation nicht hilfreich sind. Vollständigkeitsüberprüfungen sind nur in bestimmten Designsituationen sinnvoll und würden, nach jedem Schritt durchgeführt, zu einer Flut wenig hilfreicher Meldungen führen. Der Hauptkritikpunkt gegen unmittelbare Präsentation ist, daß der Designer zu sehr von der eigentlichen Aufgabe abgelenkt wird und nicht in der Lage ist, eine Folge von Designschritten ungestört auszuführen.

Um die Störung des Designprozesses möglichst gering zu halten, kann man die Informationen immer nach Abschluß der Arbeit an einer semantischen Einheit präsentieren. Wenn dies gelingt, ist es sicher ein guter Kompromiß zwischen Stören des Designprozesses und frühzeitiger Information. In mehreren Projekten hat sich jedoch gezeigt, daß es sowohl sehr schwierig ist, vorab sinnvolle semantische Einheiten zu definieren, als auch, während des Designprozesses festzustellen, wenn die Arbeit an einer dieser Einheiten abgeschlossen ist.

In VDDE [6], einem System, welches Designer bei der Erstellung von automatischen Telephoninformationssystemen unterstützt, kann eingestellt werden, daß Kritikinformationen nur dann zur Verfügung gestellt werden, wenn der Designer die Arbeit an einem neuen Teildialog aufnimmt. Dies ist jedoch dann problematisch wenn Designer an mehreren Teildialogen gleichzeitig arbeiten und zum Beispiel Unterbereiche eines Dialogs mit einem anderen Dialog austauschen wollen.

Dieses Problem wurde in FRAMER gelöst, indem der Designer durch die Verwendung der „Checklists“ dem System explizit mitteilt, wann die Arbeit an einer semantischen Einheit abgeschlossen ist [8]. So sind in FRAMER das Layout der Fensterbereiche und die Zuordnung von Input- und Output-Funktionen zu diesen Bereichen zwei unterschiedliche Designschritte, welche vom Benutzer separat behandelt und „abgehakt“ werden können, und dann auch jeweils von der Kritikkomponente separat überprüft werden.

Eine weitere Möglichkeit ist die Analyse des Designs auf Benutzeranfrage. Dieser Ansatz wird im Projekt KRI benutzt [9]. Der Vorteil des Kritik auf explizite Anforderung des Designers ist, daß auch sehr zeitaufwendige Analysen durchgeführt werden können. Dies wird jedoch damit erkauft, daß die eigentliche Stärke des Kritikansatzes, die aktive Präsentation von Informationen wenn sie benötigt werden, verloren geht.

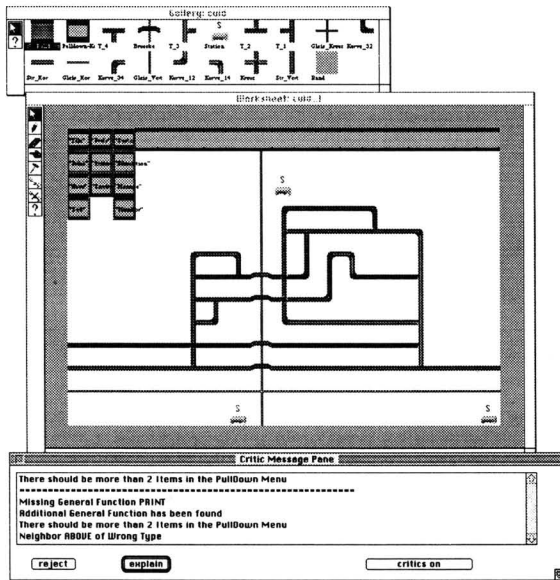


Abbildung 1: Galerie der Designobjekte, Entwurfswindow und Kritikmeldungen in CUID.

In CUID (Abbildung 1), einem Prototyp einer Kritikkomponente für den Entwurf von grafischen Benutzungsoberflächen für Verkehrsmanagementsysteme, wird dieses Problem umgangen, indem die aktive Informationspräsentation unmittelbar nach der Designaktion mit der Analyse auf Anforderung kombiniert wird. Nach jedem Designschritt werden Regeln überprüft, die sich im wesentlichen auf das Layout der Elemente in der Benutzungsoberfläche beziehen. Das System ermöglicht dem Benutzer aber auch, eine Funktion "Überprüfung des Gesamtentwurfs" auszuwählen. Mit dieser Funktion wird das gesamte Design unter Verwendung aller verfügbaren Regeln untersucht. Dazu gehören dann z.B. auch Regeln, die überprüfen, ob alle laut Spezifikation benötigten Funktionen im System vorhanden sind und ob zusätzliche, nicht geforderte Funktionen implementiert wurden. Diese Regeln werden während des Designvorgangs nicht überprüft.

2.2 Art und Ort der Informationspräsentation

Da die Art der Informationsdarstellung sehr eng mit dem Ort, an der die Information präsentiert wird, zusammenhängt, werden wir die verschiedenen Ansätze auch unter beiden Aspekten gemeinsam vorstellen. Grundsätzlich unterscheiden wir textuelle

Kritik, in der die Information als Text dargeboten wird, und grafische Kritik, bei der Information graphisch präsentiert wird.

Textuelle Kritik

Traditionell werden die Informationen von einer Kritikkomponente in einem separaten Fenster angeboten (siehe z.B. [2] für eine Übersicht). Die Variation der Art der Darstellung liegt dabei im wesentlichen in der Ausführlichkeit der Informationen. Auf der einen Seite des Spektrums steht ein Stichwort als Problembeschreibung, auf der anderen Seite eine vollständige und ausführliche Beschreibung eines Problems mit allen möglichen Ursachen und einer Erläuterung der Regeln, die dieses Problem identifizieren.

Welcher Ansatz sinnvoll ist, hängt von verschiedenen Faktoren ab. Für erfahrene Designer ist eine Liste von Stichwörtern mit den noch ausstehenden Problemen sinnvoll und oft ausreichend. Bei Bedarf muß natürlich auch für diese Designer eine ausführlichere Beschreibung zur Verfügung stehen. Für weniger erfahrene Designer wie z.B. einen Endanwender der die Benutzungsoberfläche auf seine Bedürfnisse anpaßt, ist die Präsentation einer ausführlicheren Nachricht sinnvoller.

Außerdem spielt die Menge der zu einem bestimmten Zeitpunkt präsentierten Meldungen eine entscheidende Rolle. Wenn viele Meldungen zur gleichen Zeit darzustellen sind, sind kürzere Informationen zu bevorzugen, wenn nur eine Meldung ansteht, kann diese mehr Platz in Anspruch nehmen. In jedem Fall muß der Zugriff auf weitergehende Erläuterungen zur Verfügung stehen.

FRAMER ist ein Beispiel eines Systems in welchem dem die Kritik auf textuelle Weise in einem Fenster dargestellt wird. Durch die Verwendung der Checkliste ist zu jedem Zeitpunkt die Menge der möglichen Kritikinformationen relativ gering und die Texte können daher relativ ausführlich sein. Trotz dieser Ausführlichkeit erlaubt FRAMER aber den Zugriff auf zusätzliche, erklärende Informationen.

Die separat dargestellte textuellen Kritik hat einen entscheidenden Nachteil: Der Bezug zu den von der Meldung betroffenen Objekten im Design ist sehr schwer herzustellen. Dieses Problem kann durch eine Kombination mit grafischen Ansätzen gelöst werden, die wir später diskutieren werden. Ein Vorteil der Trennung des Designs und der Kritikmeldungen ist, daß der Designer durch eine Meldung nicht unmittelbar unterbrochen wird, sondern diese erst wahrnimmt, wenn er seine Aufmerksamkeit bewußt auf das Meldungsfenster lenkt. Andererseits ist gerade dies auch wieder ein Nachteil, da damit trotz aktiver Präsentation durch das System auch ein aktives Nachschauen des Designers notwendig ist.

Eine weitere Möglichkeit der textuellen Kritik, von der uns jedoch bisher keine prototypische Anwendung im Bereich Benutzungsoberflächendesign bekannt ist, präsentiert die Informationen im Stil von "Bubble Help", wie es in Macintosh-Systemen vorhanden ist. Dabei werden die Informationen in der Form einer Sprechblase an dem betroffenen Objekt angezeigt. Dieses Verhalten der Kritikkomponente unterbricht den eigentlichen Designprozeß sehr aggressiv, ein Ignorieren der Meldung ist praktisch nicht möglich. Der Text, der in der Blase angezeigt werden kann, darf nur sehr kurz sein. Außerdem verdeckt die Sprechblase einen Teil des Designs. Trotz dieser Nachteile kann es in einer Kritikkomponente eine Verwendung für diese Art der Präsentation geben, z.B. für sehr dringende Nachrichten, die auf keinen Fall ignoriert werden dürfen.

Ein ähnlicher Ansatz ist die Präsentation der Information in einer modalen Meldungsbox. Dies ist die aggressivste Möglichkeit, den Dialog des Designers mit der Designumgebung auf die Kritikmeldungen zu zwingen.

Eine Möglichkeit der Darstellung, die weder grafische Kritik noch eigentlich textuelle Kritik ist, ist ein Flag, das anzeigt, ob in dem bearbeiteten Design Probleme existieren oder nicht. Ein derartiger Ansatz ist in [14] vorgeschlagen worden, um den Endbenutzer auf Adaptionsvorschläge des Systems hinzuweisen. Dieses Flag könnte im Rahmen des Designfensters untergebracht und damit gut sichtbar sein, wäre auf der anderen Seite leicht zu ignorieren. Auf Wunsch kann der Designer dann die Informationen abrufen, die zur Verfügung stehen. Diese können dann sowohl grafisch als auch textuell präsentiert werden.

Grafische Kritik

Grundsätzlich unterscheiden wir zwei Arten der grafischen Präsentation von Kritikinformationen: Auf der einen Seite kann das Designobjekt selbst grafisch verändert werden, andererseits kann eine Information außerhalb des Designs grafisch dargestellt werden.

Bei der ersten Möglichkeit, der Veränderung des Designobjekts selbst, ist das Problem des Bezugs zwischen einer Information und dem betroffenen Objekt ausgeräumt. Designprobleme oder bestimmte Zwischenzustände, z.B. wenn ein Objekt nicht vollständig spezifiziert ist, können durch Veränderung von Objektattributen angezeigt werden. Besonders geeignet scheinen Farbe, Intensität, Position, Markierungen und, bei Texten, Font zu sein.

Weiterhin können auch Beziehungen zwischen verschiedenen Objekten dargestellt werden, z.B. indem Linien eingezeichnet werden, die eine geometrische Relation of-

fensichtlich machen. Schließlich kann durch verschiedene Cursorformen der zu erwartende Effekt einer Designaktion visualisiert werden.

Es ist offensichtlich, daß durch die beschränkte Anzahl von grafischen Möglichkeiten auch die Menge der darstellbaren Probleme beschränkt ist. Schwerwiegender ist jedoch, daß der Designer die Bedeutung der Präsentation erkennen soll, also die Codierung lernen muß. Zum Ausgleich dafür erlaubt die graphische Darstellung von Informationen erfahrenen Designern ein schnelles und müheloses Verstehen der Kritik. Wird die Kritik im Blickfeld von erfahrenen Designern auf graphische Art dargestellt, dann kann die Information direkt und intuitiv aufgenommen werden – und bei Bedarf ignoriert werden, ohne daß der eigentliche Designprozeß unnötig unterbrochen werden muß.

Werkzeuge für das Design von Benutzungsoberflächen, welche graphische Methoden für die Bereitstellung von Kritikinformationen benutzen, sind uns nicht bekannt. Abbildung 2 vermittelt einen Eindruck, wie Küchen-Designer mittels graphischer Kritik unterstützt werden könnten. Die Spülmaschine (DW) ist in diesem Fall farbig markiert, da in der Spezifikation festgelegt ist, daß diese Küche für einen linkshändigen Benutzer entworfen werden soll. Daher sollte die Spülmaschine auf der anderen Seite des Spülbeckens stehen.

Aufgrund der Möglichkeit, Informationen direkt im Blickfeld des Benutzers anzuzeigen, eignet sich graphische Kritik besonders für die Darstellung von Informationen während und direkt nach der Benutzeraktion. Zudem, sollte für die Anzeige der Kritik während der Benutzeraktion eine Präsentation im Blickfeld des Benutzers gewählt werden damit dem Benutzer unnötige Wechsel der Blickrichtung erspart werden.

Die zweite Variante der grafischen Präsentation von Informationen ist die Darstellung außerhalb des eigentlichen Designs. Neben der Ergänzung von textuellen Erläuterungen durch Bilder und Grafiken kann es auch eine rein grafische Präsentation von Informationen sein, in unserem Beispiel (Abbildung 2) in Form von Balkendiagrammen. Diese Darstellung scheint besonders für die Veranschaulichung von sich ständig ändernden Werten geeignet. Der Designer hat damit die Möglichkeit, während oder direkt nach der Designaktion ohne großen Aufwand festzustellen, ob sich die Aktion den gewünschten Effekt bezüglich der dargestellten Werte hat.

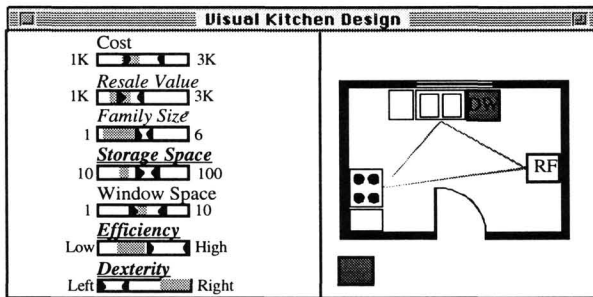


Abbildung 2: Prototyp eines Systems, welches Küchendesigner mittels graphischer Kritik unterstützt.

2.3 Kombination und Integration

Schon aus der Vorstellung der verschiedenen Möglichkeiten der Informationspräsentation wird klar, daß ein sinnvoller Weg nur mit einer Kombination beschritten werden kann. Es ist auch klar, daß der zeitliche Aspekt dabei eine entscheidende Rolle spielt. Wann eine Information präsentiert wird, hängt von zwei völlig verschiedenen und unabhängigen Faktoren ab. Der erste Faktor ist der Aufwand, den die Kritikkomponente betreiben muß, um die Information zu generieren. Wenn es nicht möglich ist, aufwendige Überprüfungen in Realzeit durchzuführen, kann diese Information auch nicht unmittelbar nach einer Designaktion angezeigt werden. Da dieser Aufwand sehr schwer abzuschätzen ist und wir uns hier auch nicht auf ein spezielles Computersystem mit einer speziellen Leistung beziehen, soll dieser Aspekt nicht weiter betrachtet werden.

Wichtiger ist die Frage nach der kognitiven Belastung des Designers: Ist der Gewinn aus der Präsentation der Information zu einem bestimmten Zeitpunkt größer als der Aufwand, den der Designer in die Auffassung der Information investieren muß.

Wenn Informationen präsentiert werden, soll diese Informationsvermittlung so erfolgen, daß der eigentliche Designprozeß möglichst wenig gestört wird. Der Designer soll auch in der Lage sein, die Informationen ggf. zu ignorieren. Auf der anderen Seite sollen die Informationen, je nach Dringlichkeit, unterschiedlich auffällig präsentiert werden. Zusätzlich sollen Informationen auch so dargestellt werden, daß sie dem Designer ohne großen zusätzlichen Aufwand zugänglich sind.

Um dies erreichen zu können, können verschiedene Modelle der Kombination von grafischer und textueller Kritik verwendet werden. Die offensichtlichste Möglichkeit ist, textuelle Informationen durch grafische Mittel, wie Bilder und Grafiken zu

ergänzen. Außerdem kann die gleiche Mitteilung als Text und als Graphik redundant dargestellt werden, um Benutzern die Möglichkeit zu geben, eine Mitteilung auf die individuell bevorzugte Art wahrzunehmen.

Zudem gibt es die Möglichkeit Text und Graphik zeitlich gestaffelt einzusetzen. So ist es z.B. möglich, daß Designer durch das Selektieren eines grafisch annotierten Objekts auf die zugehörige textuelle Kritik zugreifen können. Damit können weitere Details der Kritiknachricht oder eine exaktere Beschreibung des Problems gegeben werden. Diese Möglichkeit der Staffelung ist eine Möglichkeit, das Problem des eingeschränkten Codes bei der Annotierung von Designobjekten zu überwinden. Außerdem können textuelle Erklärungen graphisch dargestellte Kritik auch für weniger erfahrene Designer zugänglich machen und ihnen dadurch erlauben den graphischen „Code“ zu erlernen.

Aber auch die umgekehrte Folge der Informationsdarstellung kann sinnvoll sein. So können die Objekte, auf die sich eine textuelle Kritik bezieht, graphisch markiert werden. In dem VDDE-System kann der Designer auf eine im Meldungsfenster angezeigte Meldung klicken, um die von der Meldung betroffenen Objekte im Design durch Blinken grafisch hervorzuheben. Darüber hinaus können Kritikinformationen durch grafische Demonstrationen oder Simulationen erläutert werden. Zum Beispiel in PetriNED [12], einer Designumgebung für Petri-Netze, welche graphische Methoden der Kritikdarstellung benutzt, können, ausgehend von einer textuellen Kritik in einem Fenster, Simulationen von Deadlock-Situationen abgerufen werden (siehe Abbildung 3).

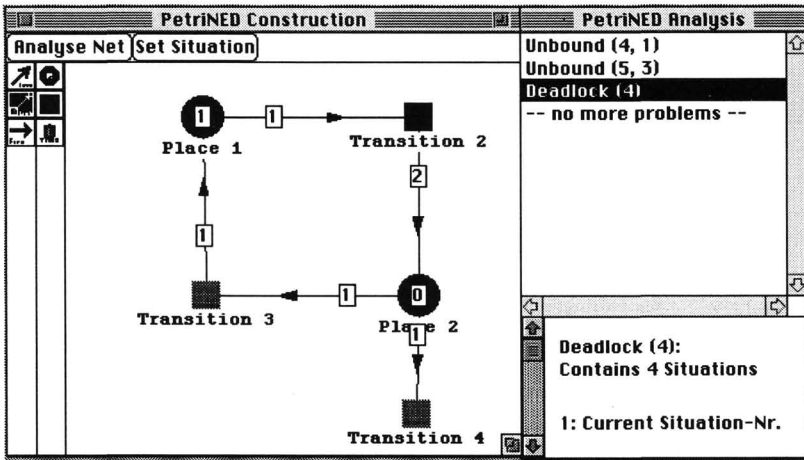


Abbildung 3: Kombination von textueller Kritik (Stichworte oben rechts) und graphischer Erklärung (Simulation im Design, Protokoll unten rechts) in PetriNED.

3 Diskussion und Ausblick

Obwohl es aus den Erfahrungen mit den bisherigen Kritiksyste men offensichtlich ist, daß eine Kombination der verschiedenen Arten der Informationspräsentation notwendig und sinnvoll ist, gibt es hierzu bisher wenig Realisierungen. Im VDDE-System kann der Designer den Bezug zwischen einer Meldung und den betroffenen Objekten durch Klicken auf die Meldung darstellen lassen; in PetriNED sind erste Erfahrungen mit der Präsentation von Simulationen und Demonstrationen von möglichen Problemen gemacht worden. Diese erlauben dem Designer, mehr Informationen über das Systemverhalten zum Ablaufzeitpunkt bereits zum Designzeitpunkt zu berücksichtigen.

Alle anderen dargestellten Aspekte der Kombination der Darstellungsarten und des Zeitpunkts sind bisher nur theoretisch betrachtet worden. In den nächsten Schritten sollen Prototypen gebaut beziehungsweise so verfeinert werden, daß die Aspekte in ausführlichen Benutzertests untersucht werden können. Es ist offensichtlich, daß nicht intuitiv festzulegen ist, wie kompliziert eine Kodierung für die grafische Präsentation von Informationen sein darf, und entsprechend, welche Informationen erst auf Anfrage textuell zur Verfügung gestellt werden sollten.

Neben der Integration von verschiedenen Techniken sollten auch Möglichkeiten für deren Individualisierbarkeit untersucht werden. So sollte der Designer einstellen

können, ob er zunächst nur ein Flag, eine grafische Annotation, eine textuelle Kritik in einem separaten Fenster, oder ein modales Meldungsfenster angezeigt bekommen möchte. Diese Einstellungsmöglichkeiten sollen auch so ausgelegt sein, daß diese Auswahl für verschiedene Arten von Meldungen unterschiedlich sein können. Bei der anschließenden Navigation durch die ergänzenden Informationen soll der Designer ebenso die Freiheit haben, die Informationsdarstellung, die ihm am angenehmsten ist, zu wählen.

Ein weitere Forschungsrichtung ist die Untersuchung von Möglichkeiten der Integration von Problemdetektion und Problembehandlung. Wie bereits diskutiert konzentrieren sich Kritiksysteme traditionellerweise auf die Problemdetektion und überlassen dem Designer die Behandlung des Problems. Unserer Meinung nach sollten Kritik-Systeme einen Schritt weiter gehen und nicht nur Probleme aufzeigen, sondern „konstruktive Kritik“ geben und den Designer aktiv bei der Behandlung von Problemen unterstützen. Ein relativ einfaches Beispiel der Integration von Problemdetektion und Problembehandlung existiert in PetriNED. PetriNED macht Designer auf mögliche positive Eigenschaften in ihrem Design mittel graphischer Kritik aufmerksam. Wenn zwei Elemente im Design ungefähr ausgerichtet sind, so wird diese positive Eigenschaft dem Designer durch das Zeichnen einer Linie zwischen den beiden Elementen mitgeteilt. Beendet der Designer seine Aktion während eine solche Linie im Design erscheint, dann stellt das System sicher, daß die Elemente genau ausgerichtet werden. Im Benutzungsoberflächen-Design könnte ein Kritik-System welches „konstruktive Kritik“ gibt dem Designer zum Beispiel eine Auswahl von sinnvollen Lösungsmöglichkeiten geben. Weitere Möglichkeiten und Techniken der konstruktiven Kritik für die Unterstützung von Benutzungsoberflächen-Designern sollten exploriert werden.

Literatur

- [1] Barber, J., S. Bhatta, A. Goel, M. Jacobson, M. Pearce, L. Penberthy, M. Shankar, R. Simpson & E. Stroulia (1992). AskJef: Integration of Case-Based and Multimedia Technologies for Interface Design Support. In J. S. Gero (Ed.) *Artificial Intelligence in Design '92*. Amsterdam, NL: Kluwer Academic Publishers. 457-475.
- [2] Fischer, G., A. C. Lemke, T. Mastaglio & A. I. Morch (1991). Critics: An Emerging Approach to Knowledge-Based Human Computer Interaction. *International Journal of Man-Machine Studies* 35(5): 695-721.
- [3] Fischer, G., R. McCall & A. Morch (1989). Design Environments for Constructive and Argumentative Design. In (Ed.) *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*. 269-275.
- [4] Foley, J., W. C. Kim, S. Kovacevic & K. Murray (1989). Defining User Interfaces at a High Level of Abstraction. *IEEE Software* 6(1): 25-32.

- [5] Garfinkel, S. L. & M. K. Mahoney (1993). *NeXTSTEP Programming. Step One: Object-Oriented Applications*. New York: Springer-Verlag.
- [6] Harstad, B., K. Nakakoji & T. Sumner (1994). Perspective-Based Critiquing: Helping Designers Cope with Conflicts Among Design Intentions. In J. S. Gero (Ed.) *Artificial Intelligence in Design '94*. Dordrecht: Kluwer.
- [7] Kühme, T. & M. Schneider-Hufschmidt (1992). SX/Tools - An Open Design Environment for Adaptable Multimedia Interfaces. *Proc. Eurographics '92, Cambridge UK, Sept 7-11, 92*.
- [8] Lemke, A. C. & G. Fischer (1990). A Cooperative Problem Solving System for User Interface Design. In (Ed.) *Proceedings of AAAI-90, Eighth National Conference on Artificial Intelligence*. Cambridge, MA: AAAI Press/The MIT Press. 479-484.
- [9] Löwgren, J. & U. Laurén (1993). Supporting the use of guidelines and styleguides in professional user interface design. *Interacting with Computers* 5(4): 385 - 396.
- [10] Luo, P., P. Szekely & R. Neches (1993). Management of Interface Design in HUMANOID. In (Ed.) *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*. 107-114.
- [11] Reiterer, H. (1993). Human Factors Based Interface Design. In T. Grechening and M. Tscheligi (Ed.) *Human Computer Interaction, Vienna Conference VHCI '93*. Wien: Springer-Verlag. 291-302.
- [12] Stolze, M. (1994). Visual critiquing in domain oriented design environments: Showing the right thing at the right place, in Gero, J. S. (ed.), *Artificial Intelligence in Design '94*, Kluwer, Dordrecht.
- [13] Sukaviriya, P., J. D. Foley & T. Griffith (1993). A Second Generation User Interface Design Environment: The Model and the Runtime Architecture. In (Ed.) *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*. 375-382.
- [14] Thomas, C. G. & M. Krogsoeter (1993). An Adaptive Environment for the User Interface of Excel. In (Ed.) *Proceedings of the 1993 International Workshop on Intelligent User Interfaces*. 123-130.

Uwe Malinowski

Siemens Zentralabteilung Forschung und Entwicklung (ZFE T SN 5)

D 81730 München

Uwe.Malinowski@zfe.siemens.de

Markus Stolze

Department of Computer Science, University of Colorado at Boulder

Boulder, CO 80309-0430, USA

stolze@cs.colorado.edu

