Style Guides am Beispiel der „Java Look and Feel Design Guidelines“ von Sun Microsystems - eine kritische Betrachtung

Roland Petrasch
FH Nordakademie, Elmshorn
petrasch@nordakademie.de

Abstract

Die „Java Look and Feel Design Guidelines“ von Sun Microsystems liefern Vorgaben für die Entwicklung von graphischen Benutzungsschnittstellen, die auf der Basis der Java Foundation Classes entstehen. Mit diesen Werk versucht Sun eine Lücke zu schließen, die durch die Verbreitung von Java und dem Fehlen einer Richtlinie vom Erfinder dieser plattformübergreifenden Sprache entstanden war. Dieser Beitrag betrachtet die im Style Guide dargelegten Anforderungen an das User Interface von Java-Applikationen und gibt einige kritische Anmerkungen.

1 Aufgaben und Gültigkeit von Style Guides

Style Guides (Richtlinien) für das User Interface sind durch die Formulierung von Anforderungen an ein Produkt, eine Produktkategorie bzw. dessen Komponenten ein wichtiges Instrument der konstruktiven Qualitätssicherung, das sich auch für analytische Techniken einsetzen läßt, z.B. statische Prüfungen durch Checklisten [12]. Style Guides für spezielle Produkte entstehen i.d.R. nicht durch ein Normungs[1]- bzw. Standardisierungsverfahren[2], sondern durch privatwirtschaftliche Vereinigungen, z.B. der Motif Style Guide (The Open Group) [16], oder Einzelunternehmen, z.B. Microsoft’s Windows Interface Guidelines for Software Design [10]. Motivation für die Schaffung von Style Guides ist oftmals die starke Marktdurchdringung des Produktes.

Während Normen eine hohen Grad an Allgemeingültigkeit aufweisen, dem Nutzen der Allgemeinheit dienen sowie unabhängig von einem Produkt sein wollen, vermitteln Style Guides eher produktspezifische Vorgaben[3]. Der Aspekt der Vereinheitlichung kommt in jedem Fall zu Tragen, wobei Normen für die Qualität von Software-Systemen mit der Schwierigkeit umgehen müssen, daß es „nur wenig allgemein akzeptierte Verfahren zur Bestimmung von Softwarequalität gibt“ [9]. Auch müssen Normenwerke häufig erst interpretiert bzw. operationalisiert werden, damit sie in der Praxis zum Tragen kommen[4]. Zur Problematik der Umsetzbarkeit kommt die Vielzahl an Normen wie z.B. die ISO 9241-10 [5], die ergonomische Anforderungen aufführt, die ISO 9126 [6] mit ihren allgemeinen Qualitätsmerkmalen oder die ISO 12119 [7], welche Qualitätsanforderungen und Prüfbestimmungen vorgibt[5].

---

[1] Die DIN 820 definiert Normung wie folgt: "Normung ist die planmäßige, durch die interessierten Kreise gemeinschaftlich durchgeführte Vereinheitlichung von materiellen und immateriellen Gegenständen zum Nutzen der Allgemeinheit." [8]

[2] Der ISO/IEC Guide 2 definiert den Begriff „standard“ „as a document, established by consensus and approved by a recognized body, that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context.“ [4]

[3] Es ist hierbei zu beachten, daß sich die Begriffe „Norm“, „Standard“ und „Richtlinie“ terminologisch nicht klar abgrenzen lassen, da beispielsweise in einigen Normenwerken auch von „guidelines“ gesprochen wird.

[4] Ein Beispiel sei mit der ISO 9241-10 gegeben: „ ... these principles can be applied, but as general guidelines only. The manner in which each dialogue principle can be applied will depend on the characteristics of the intended user of the system, the tasks, the environment and the specific dialogue technique used.“

[5] Besonders die Verwendung der Terminologie im Themenbereich „Software-Qualität“ ist bei vielen Normen noch unbefriedigend, so spricht z.B. die ISO 9141 von „requirements“, während es bei der ISO 9126 um „characteristics geht. Diese Kritik ist auch bei den prozeßorientierten Normen und Standards angebracht [13].

Den existierenden produktorientierten Style Guides ist gemein, daß sie nicht oder nur kurz auf die Grundprinzipien der Software-Ergonomie und des Software-Entwurfes verweisen und entsprechende Literatur sowie Normen nennen. Gefordert werden z.B. häufig die Verwendung von Metaphern, die Konsistenz bei den Interaktionsmechanismen und die Bereitstellung von kontextspezifischen Hilfetexten. Der Schwerpunkt liegt jedoch eindeutig bei der Beschreibung der Interaktionselemente eines User Interface und deren Verhalten beim Eintreten von Interaktionen sowie der Formulierung von Vorgaben bei der Gestaltung des User Interface. Durch ihre Produktnähe zeigt sich der praktische Nutzen solcher Richtlinien bei der Entwicklung recht schnell und unmittelbar. Die Ansprüche an wissenschaftliche Literatur wollen diese Werke i.d.R. nicht erfüllen und sind daher als praxisnahe Leitfäden zu verstehen.

Ein Ineinandergreifen von Normen und Produktrichtlinien, die es dem Anwender ermöglicht, z.B. die Prinzipien der Software-Ergonomie zu verstehen und diese nachvollziehbar mit einem Style Guide anzuwenden, ist nicht zu beobachten. Auch die wissenschaftliche Literatur kann dieses Defizit nur teilweise beheben, obwohl es positive Beispiele gibt, z.B. im Bereich „ergonomische Evaluierung“ mit dem ERGOguide von Dzida et. al. [2].

2 Struktur und Inhalt der „Java Look and Feel Design Guidelines“

2.1 Überblick

Das Werk ist sowohl frei im Internet[6] als auch in Buchform [15] erhältlich. Es umfaßt ca. 230 Seiten und drei unterteilt sich in drei Teile: Teil 1 erklärt das Java Look & Feel (L&F) und führt in die JFC ein, während sich Teil 2 Entwurfsentscheidungen für Java-Anwendungen, allgemeinen bzw. grundlegenden Themen der visuellen Gestaltung und dem Verhalten von Interaktionselementen widmet. Teil 3 schließlich legt in 6 Kapiteln ausführlich die Komponenten der JFC dar und stellt Anforderungen an deren Gestaltung und Einsatz. Ein Glossar und ein Anhang über die Handhabung des UI mit der Tastatur ergänzen die Richtlinie.

2.2 Inhalt der Richtlinie und kritische Anmerkungen

2.2.1 Allgemeines

Zunächst sei erwähnt, daß Kritik auch positiv sein kann (was leider allzu häufig vergessen wird), so daß hier natürlich auch lobenswerte Punkte aufgeführt sind. Die redaktionelle Gestaltung und die strukturelle Aufteilung der Richtlinie verdienen eine solche positive Erwähnung, da Übersichtlichkeit, Verständlichkeit und Gestaltungsattraktivität gut der mit der komplexen und umfangreichen Materie in Einklang gebracht wurden.

2.2.2 Einführung

Die Einführung klärt über den angesprochenen Leserkreis auf: Der Designer des User Interface ist der Adressat, d.h. die mit dem Layout und der Auswahl von Interaktionselementen betraute Person. Aber auch für andere Personenkreise wie Marketingspezialisten, Qualitätsbeauftragte oder technische Autoren könnten die Ausführungen von Interesse sein. Technische Aspekte bzgl. der Implementierung im Java, die für Software-Entwickler wichtig sind, werden nicht behandelt. Es wird in diesem Zusammenhang auf die Web-Seiten für Java und JFC/Swing verwiesen.

Einschränkungen in Hinblick auf den Gültigkeitsbereich führt die Richtlinie ebenfalls auf: Gemäß dem Namen „Java L&F Design Guidelines“ betrachtet sie ausschließlich Applets und Applikationen auf der Basis der Java Foundation Classes für den PC bzw. NC. Damit grenzt sich das Werk gegen eingebettete Systeme (z.B. Anwendungen für Handy, PDAs etc.) und andere Technologien ab, z.B. HTML, XML, Java Script, die keine Berücksichtigung finden[7].

---

[6] Es ist unter http://java.sun.com/products/jlf/dg/index.htm zu finden.

[7] Dem drohenden Fehler, die „Java Look and Feel Design Guidelines“ als Richtlinie für Internet- oder Web-Applikationen zu bezeichnen, sei hiermit entgegengewirkt.

Auch will das Werk weder Prinzipien aus dem Bereich Software-Ergonomie, z.B. die Wahrnehmungsgesetze [14], erläutern, noch die Qualitätssicherung, z.B. für Usability Testing, direkt ansprechen[8]. Somit findet sich auch keine Checkliste für die Prüfung auf Konformität zur Richtlinie wie dies z.B. im Motif Style Guide [16] der Fall ist.

Graphische Symbole für die Klassifizierung von Textpassagen, z.B. (Anforderungen an das L&F), (Implementationshinweise) und (Cross-Plattformempfehlungen) sind im Wortwort definiert. Ihre Verwendung ist hilfreich und findet sich in allen Teilen der Richtlinie.

2.2.3 Teil 1

Der Teil 1 düfte für Entwickler und Designer, die mit Java-Applikationen vertraut sind, wenig interessant sein, bietet jedoch dem Java-Neuling einen äußerst hilfreichen Einstieg: Kapitel 1 zeigt das Java Look and Feel, z.B."flush 3D style, „drag texture, und „color model". Auch Beispiele finden sich hier. Kapitel 2 führt die JFC in Zusammenhang mit dem Java 2 SDK ein und listet die am häufigsten anzutreffenden Komponenten der JFC auf. Begriffe wie Ein- und Ausgabehilfen für spezielle Benutzergruppen (Accessibility) und die Internationalisierung werden eingeführt.

2.2.4 Teil 2

Dieser Teil besteht aus 4 Kapiteln: Kapitel 3 beginnt mit Entwurfentscheidungen. In knapper Form kommen Bereiche wie Java-Applets und -Applikationen, Bereitstellung der Anwendung und Sicherheit zur Sprache. Etwas ausführlicher fallen Erläuterungen zu den Themen Internationalisierung und Accessibility aus- Letzteres sind Ein- und Ausgabehilfen für Anwender mit speziellen Anforderungen, z.B. Mnemonics für „Power User" oder Benutzer mit einer Behinderung[9]. Kapitel 4 befaßt sich dann mit den Grundlagen der visuellen Gestaltung, d.h. Farben, Schriftarten, Textattributen, Layoutaspekten wie Abständen und Gitter. Kurz wird auch auf Animation eingegangen. Der Konkretisierungsgrad der Forderungen ist unterschiedlich. So wird z.B. bei Farben darauf verweisen, daß die Primärfarbe „genug" Kontrast zur Hintergrundfarbe haben sollte (ohne dies näher zu spezifizieren), während sich bei Abständen zwischen Elementen die exakte Angabe von 5 Pixel findet. Obwohl auch allgemeine Aussagen für die ergonomische Gestaltung vorhanden sind, z.B. „Use language that is clear, consistent, and concise throughout your application text..., fehlen wichtige Forderungspunkte, z.B. „keine hochgesättigten, spektral gegensätzlichen Farben wie Rot und Blau gleichzeitig darbieten" [1].

Kapitel 5 beschreibt Graphikelemente, indem es auf Farben im Cross-Plattformumfeld und auf die Gestaltung von Buttongraphiken, Icons und Symbolen eingeht. Dies erfolgt außer für Symbole recht umfassend. Der Punkt „Designing Graphics for Corporate and Product Identity" fällt dagegen eher enttäuschend aus. Kapitel 6 stellt das Verhalten von Eingabegeräten und Interaktionselementen vor: Mausoperationen, Mauszeigertypen, Tastaturnavigation und –aktivierung, Shortcuts sowie Mnemonics werden abgehandelt, wobei viele Punkte als Erläuterung zu verstehen sind und weniger Forderungen darstellen, z.B. die „Pointer Types" oder die „Common Navigation and Activation Keys".

2.2.5 Teil 3

Teil 3 umfaßt die Kapitel für die ausführliche Beschreibung der Komponenten der JFC. Kapitel 7 beginnt mit Windows, Panes und Frames. Es finden sich zahlreiche Bilder mit Beispielen und Erläuterungen. Die Forderungen sind weitestgehend verständlich und nachvoll-

[8] Es finden sich umfangreiche Literatur angaben über „Design Principles" im allgemeinen und für spezielle Plattformen.

[9] Dieser Bereich ist sicherlich noch eingehender vom Designer zu behandeln. Letztendlich können viele der in der Richtlinie genannten Forderungen nur für dann gelten, wenn Benutzer ohne Behinderungen vorausgesetzt werden. So führt beispielsweise das Kapitel 4 – Visual Design auf, daß der Abstand zwischen zusammengehörigen Interaktionselementen 5 Pixel betragen soll, was bei einigen Benutzergruppen sicherlich nicht ausreichend ist.

ziehbar, auch wenn sie teilweise selbstverständliche Aussagen machen, z.B. „Display a horizontal scrollbar if the view cannot show everything...“. Unbefriedigend sind die Forderung bzgl. des Utility Windows, welches i.d.R. eine „collection of tools, colors, or patterns“ zeigt. Hier heißt es, daß ein solches Window vom Primär-Window unabhängig ist und daher bei dessen Schließen nicht ebenfalls geschlossen werden soll[10]. Ein Utility Window ohne Primär-Window dürfte in dem meisten Fällen aus Sicht des Anwenders wenig Sinn machen. Auch ist nicht nachvollziehbar, warum ein Utility-Window nicht unabhängig vom Fokus im Vordergrund sein soll[11].

Kapitel 8 widmet sich Dialogen. Auch hier finden sich ausführliche und hilfreiche Erläuterungen, allerdings auch einige verbesserungswürdige Punkte, z.B. redundante Ausführungen[12]. Auch die Beschreibung zur Hilfe sind nicht besonders überzeugend. Ein Unterkapitel „Help Buttons“ gibt es zwar, bleibt jedoch recht unverbindlich, z.B. „You can use a Help button in any dialog box ...“[13]. Der Punkt „OK and Cancel Buttons“ erwähnt zwar einen Hilfe-Button, macht hierfür jedoch keine weiteren Vorgaben. Unter der Überschrift „Alert Boxes“ erläutert die Richtlinie einige Typen, z.B. „Info Alert Boxes“ oder „Warning Alert Boxes“, wobei fraglich ist, ob die Bezeichnung „Question Alert Boxes“ adäquat ist, da eine Nachfrage im Sinne des Feedbacks wohl kaum als Alarmzustand zu charakterisieren ist[14].

Das bisher gesagte gilt im Prinzip auch für Kapitel 9 bis 12, die Menüs, Toolbars und elementare sowie komplexe Interaktionselemente beschreiben. Viele sinnvolle und praxisnahe Anmerkungen und Vorgaben finden sich, könnten jedoch noch Ergänzungen erfahren, z.B. bliebt offen, ob ein Radio-Buttons, die gemeinsam eine Optionsgruppe („radio set“) bilden, durch eine Groupbox optisch gruppiert werden sollten. Auch Entscheidungshilfen, wann Elemente wie Radio Button, Combo-Box oder Text Field eingesetzt werden gibt, sind unvollständig, z.B. max. 9 Optionen bei Radio Buttons. Fragwürdig sind einige Kombinationen aus Interaktionselementen, z.B. Radio Button Menu Items.

3 Fazit und Ausblick

Insgesamt sind die „Java Look and Feel Design Guidelines“ ein für die definierte Zielgruppe gelungenes Werk, welches zwar umfangreich, der Thematik jedoch sicherlich angemessen ist. Immerhin handelt es sich um die Version 1.02, so daß die Evolution der Richtlinie gerade begonnen haben dürfte. Durch die Berücksichtigung zahlreicher Aspekte ist es für den User Interface Designer ein hilfreiches Werk. Die Vorgaben und Forderungen könnten allerdings noch sinnvoll ergänzt werden, z.B. für die Farbgestaltung. Auch wenn keine Prinzipien behandelt werden sollen, wäre z.B. die Erwähnung der 7+-Regel angebracht [11][15].

Um tatsächlich anwendergerechte Java-Anwendungen entwickeln zu können, sind zusätzlich Kenntnisse im Bereich „Software-Ergonomie“, z.B. Gestaltungsprinzipien, Wahrnehmungsgesetze etc., notwendig. Hier liegt ein Problempotential: Die Richtlinie will und kann die wissenschaftliche Literatur nicht ersetzen, sondern verweist explizit auf solche Werke, d.h. nur die Beachtung der „Java Look and Feel Design Guidelines“ allein sichert noch nicht die Entstehung ergonomischer Software. Ein Gütesiegel, welches Konformität zu dieser Richtlinie bescheinigt (ähnlich dem Label „100% Pure Java“) wäre sicherlich irreführend und erscheint nicht empfehlenswert. Dennoch ist die Schaffung der hier betrachteten Richtlinie als

---

[10] Für ein Sekundär-Window gilt dies nicht: „Keep track of the secondary windows in your application; close them if the primary window is closed or hide them if their primary window is minimized.“

[11] Hier heißt es: „... utility windows do not float above all the other windows...“. Das würde bedeuten, daß das Primär-Window beim Fokuserhalt das Utility Windows überdecken würde, was i.d.R. nicht wünschenswert ist.

[12] Unter „Dialog Box Design“ heißt es: „Include mnemonics for all user interface elements except the default button and the Cancel button“. Bei der Beschreibung für „OK and Cancel Buttons“ findet sich die Vorgabe „Do not add a mnemonic to the Cancel button.“.

[13] Hier wären Themen wie kontextsensitive Hilfe, Hilfe-Dialoge, Hilfe-Assistenten und andere Mechanismen wie „Mouse-Over“-Effekt auch an anderer Stelle sinnvoll gewesen.

[14] Als Beispiel für eine „Question Alert Box“ wird ein Dialog für die Eingabe eines Namens aufgeführt.

[15] Evtl. wird der Maximalwert von 9 „Chunks“ sogar noch herabgesetzt werden müssen [3].

richtig zu erachten und zu hoffen, daß diese wie die plattformunabhängige Sprache Java selbst eine Breite praktische Relevanz erhält und unter Beteiligung der Java Community weiterentwickelt wird.

Allerdings sollten Style Guides wie die „Java Look and Feel Design Guidelines“ besser mit Normen und Standards in Einklang gebracht werden. Hierzu sind jedoch noch einige Weiterentwicklungen notwendig, insbesondere die Harmonisierung der Terminologie könnte verbessert werden, damit Normen wie die ISO 9241-10, die ISO/IEC 9126 oder die ISO/IEC 12119 in Verbindung mit Produktrichtlinien zum Einsatz kommen. Verbesserungen und Neuentwicklungen von Normen sind daher zu erwarten, z.B. die ISO 14915 Multimedia User Interface Design[16].

[1] Balzert, H.: *Lehrbuch der Software-Technik - Software-Entwicklung*. Spektrum Akademischer Verlag Heidelberg Berlin Oxford, 1996

[2] Dzida, W.; Wiethoff, M.; Arnold, A.: *ERGOguide – The Quality Assurance Guide to Ergonomic Software*. Delft University, The Netherlands & GMD, Germany, 1993

[3] Glaser, W. R.: *Menschliche Informationsverarbeitung*. In: Eberleh, E., Oberquelle, H., Oppermann, R. (Hrsg.): Einführung in die Software-Ergonomie. 2. Auflage, Walter de Gruyter, Berlin, New York, 1994

[4] ISO/IEC Guide 2:1996: *Standardization and related activities -- General vocabulary*, ISO, 1996

[5] ISO 9241-10: *Ergonomic Requirements for Office Work with VDTs*. Part 10: *Dialog Principles*. ISO, 1996

[6] ISO/IEC 9126: *Quality Characteristics and Guidelines for Their Use*. ISO/IEC, 1991

[7] ISO/IEC 12119: *Information technology – Softare Packages; Quality Requirements and Testing*. ISO / IEC, Nov. 1994

[8] DIN 820-1: *Normingsarbeit, Grundsätze*. Beuth Verlag, Berlin 1994

[9] Lindermeier, R.; Siebert, E.: *Softwareprüfung und Qualitätssicherung*. 2. Auflage R. Oldenbourg Verlag München, 1995

[10] Microsoft: *The Windows Interface Guidelines for Software Design : An Application Design Guide*. Mircosoft Press, 1995

[11] Miller, G.A.: *The magical number seven, plus or minus two: Some limits on our capacity for processing information*. In: Pychological Reviews, 63, 1956

[12] Petrasch, R.: *Einführung in das Software-Qualitätsmanagement*. Logos Verlag Berlin, 1998

[13] Petrasch, R.: *Über den Software-Qualitätsbegriff*. In: Softwaretechnik-Trends, Mitteilungen der GI, Nov. 1999

[14] Preece, J., Rogers, Y., Sharp, H.; Benyon, D.; Holland, S.; Carey, T.: *Human-Computer Interaction*. Addison Wesley Wokingam, Reading, Menlo Park, New York u.a., 1994

[15] Sun Microsystems: *Java Look and Feel Design Guidelines.* Addison Wesley, 1999

[16] The Open Group: *CDE 2.1/Motif 2.1*
Style Guide and Glossary
Style Guide Reference
Style Guide Certification Check List, 1997

[17] Wallmüller, E.: *Software-Qualitätssicherung in der Praxis*. Carl Hanser, München Wien, 1990

[16] Offiziell wird diese Norm von der ISO nicht aufgeführt, da sie sich offensichtlich noch im Entwurfsstadium befindet. Allerdings erwähnt die Web-Seite für „Ease of Use“ der IBM (http://www.ibm.com/ibm/easy) die Norm bereits.