

Eine Design Pattern-Sprache für mobile Applikationen mit dem Schwerpunkt Navigationssysteme

Rainer Gibbert

Amselweg 4, D-56856 Zell-Mosel

[Rainer.Gibbert\(at\)gmx.de](mailto:Rainer.Gibbert(at)gmx.de)

Zusammenfassung

Die Entwicklung und Gestaltung von Benutzerschnittstellen für mobile Applikationen stellt eine besondere Herausforderung an den Designer dar. Mobile Endgeräte wie beispielsweise PDAs, Pager, Mobiltelefone und Smartphones bringen eine Menge von Einschränkungen bezüglich ihrer Bildschirmgröße, der Eingabemöglichkeiten, der Rechenleistung usw. mit sich, die ein Designer bei der Gestaltung von Applikationen für diese Geräte mit einbeziehen muss.

Um jedoch nicht für jedes Projekt das Rad neu erfinden zu müssen, sollen einmal entdeckte Design-Lösungen gespeichert und Projektteams auf eine einfache und verständliche Weise zugänglich gemacht werden. Hierzu gibt es mehrere Ansätze, von denen jener der Design Patterns in diesem Beitrag detailliert erläutert werden soll.

Am Beispiel einer Design Pattern-Sprache für mobile Applikationen und einer speziellen Sprache für Navigationssysteme auf mobilen Endgeräten wird gezeigt, wie Design-Wissen dieser Anwendungsbereiche mit Hilfe von Design Patterns gesammelt, gespeichert und somit für zukünftige ähnliche Projekte zugänglich gemacht werden kann.

1. Einleitung

Ursprünglich von Christopher Alexander in den 1970er Jahren für architektonische Gestaltungsprobleme entwickelt [1, 2] und 1987 im Software Engineering aufgegriffen [3], wird der Design Pattern-Ansatz in den letzten Jahren auch für das User Interface- und Interaction Design als viel versprechende Methode behandelt [4]. Ziel ist es, mit gut strukturierten und interdisziplinär verständlichen Beschreibungen bewährter „Muster-Lösungen“ gesammeltes Design-Wissen in verständlicher und allgemein gültiger Form zu katalogisieren. Ein einzelnes Design Pattern beschreibt eine Lösung eines wiederkehrenden Design-Problems und gibt Beispiele für konkrete Umsetzungen. Eine Sammlung von Design-Patterns wird in einer Sprache strukturiert, d.h. einer hierarchischen Struktur von allgemeinen, abstrakten Problemen bis zu konkreten, spezifischen Designfragen.

In diesem Beitrag wird ein Beispiel für eine Design-Pattern-Sprache vorgestellt, welche als Diplomarbeit im Rahmen des Verbundprojekts SAiMotion (Situation Awareness in Motion) der Fraunhofer-Gesellschaft entstanden ist. In dem vom BMBF geförderten Projekt wurde ein mobiles Informationssystem entwickelt, welches den Besucher in einer komplexen Umgebung personalisierte und situativ angepasste Informationen bereitstellt. Als ein Anwendungsszenario unterstützt es die Vorbereitung, Durchführung und Auswertung von Messebesuchen. In SAiMotion wurden umfangreiche Studien und Usability-Tests für die Entwicklung benutzergerechter Kartendarstellungen auf mobilen Endgeräten durchgeführt. Ziel der Entwicklung der Design-Pattern-Sprache war es, dieses ergonomische Wissen integriert zu sammeln und aufzubereiten.

Neben den Design Patterns für das Anwendungsgebiet der Navigationssysteme wurde eine abstrakt gehaltene Design Pattern-Sprache entwickelt, welche Richtlinien für die Gestaltung mobiler Applikationen im Allgemeinen bietet. Insgesamt wurden so zwei Design Pattern-Sprachen erarbeitet, die zwar keine Ansprüche auf Vollständigkeit erheben, aber dennoch als Unterstützung bei der Entwicklung ähnlicher Systeme dienen zu können.

Im Folgenden wird die Struktur der Design-Pattern-Sprachen sowie konkrete Beispiele für Design Patterns vorgestellt und abschließend die Relevanz des Ansatzes für den Einsatz in Entwicklungsprojekten diskutiert.

2. Design Patterns für mobile Applikationen

Als Grundlage der Design Patterns für mobile Applikationen in diesem Beitrag dienen die Design Patterns von Noble und Weir [5]. Diese sind in einer Hierarchie angeordnet, die anhand eines zentralen Gestaltungsproblems mobiler Geräte aufgebaut ist, der beschränkten Displaygröße.

Die im Folgenden vorgestellte Struktur erweitert die Pattern-Sprache von Noble und Weir [5] um einige Patterns und modifiziert wo notwendig Darstellung und Struktur, um insgesamt eine konsistente Hierarchie zu erreichen.

Ausgangspunkt der Sprache ist das allgemeine Pattern „Mobile Application“ (vgl. Abb. 1), das Charakteristiken mobiler Geräte beschreibt, wie z.B. deren unterschiedliche Nutzungskontexte, die kleinen Displays und die beschränkten Eingabemöglichkeiten, sowie die daraus resultierenden Anforderungen, welche sich für das User Interface Design, die Informationsarchitektur und das Interaktionsdesign ergeben.

In der Hierarchie folgen zunächst Patterns zur Gestaltung von Fensteroberflächen auf mobilen Geräten (vgl. Abb. 1). Das Pattern „One True Window“ [5] beschreibt die Richtlinie, Applikationen für mobile Geräte in nur einem sichtbaren Fenster zu gestalten – im Gegensatz zu Desktop-Computern, auf denen mehrere verschiedene Anwendungen gleichzeitig dargestellt werden können. Ausnahmen zu diesem Pattern werden im referenzierten Pattern „Secondary Window“ gezeigt, welches beispielsweise für Dialoge oder Meldungen angewandt werden kann.

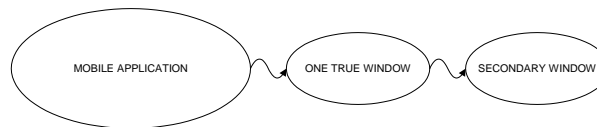


Abb. 1: Das Haupt-Pattern „Mobile Application“ sowie Patterns zum Anwendungsfenster und Sekundärfenstern

Weitere Unterpatterns zu „Mobile Application“ zeigen, wie in mobile Anwendungen Platz auf dem Display gespart werden kann, indem beispielsweise Interaktionselemente oder weitere Informationen in Tooltips oder Kontextmenüs versteckt werden können und wie Bedienelemente dargestellt und dimensioniert werden sollten (vgl. Abb. 2).

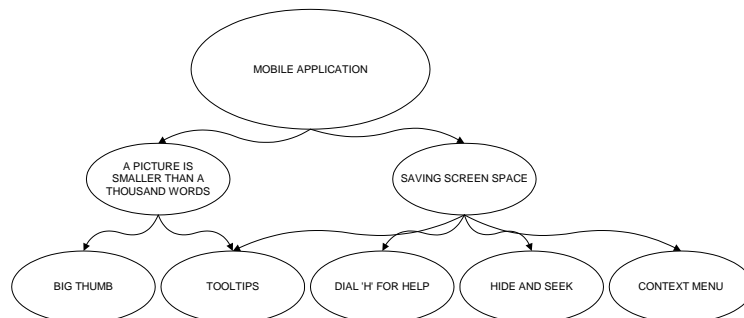


Abb. 2: Patterns zum Sparen von Platz auf dem Display und zu Gestaltung und Dimensionierung von Interaktionselementen

Patterns zur Informationsarchitektur und zur Navigationsgestaltung beschreiben, wie zum Beispiel Dialoge strukturiert werden können, komplexe Informationen in hierarchischen Strukturen dargestellt oder in Kategorien unterteilt werden sollten (vgl. Abb. 3).

„Cup Of Tea Test“ [5] wiederum beschreibt, wie mobile Applikationen so gestaltet werden können, dass Anwender die Interaktion kurzfristig oder kurzzeitig unterbrechen können, ohne dass Datenverluste entstehen oder die Orientierung im Interaktionsablauf verloren geht. Als Lösung sollten beispielsweise immer Informationen zum aktuellen Status angezeigt oder in einen sicheren Zustand zurückgekehrt werden.

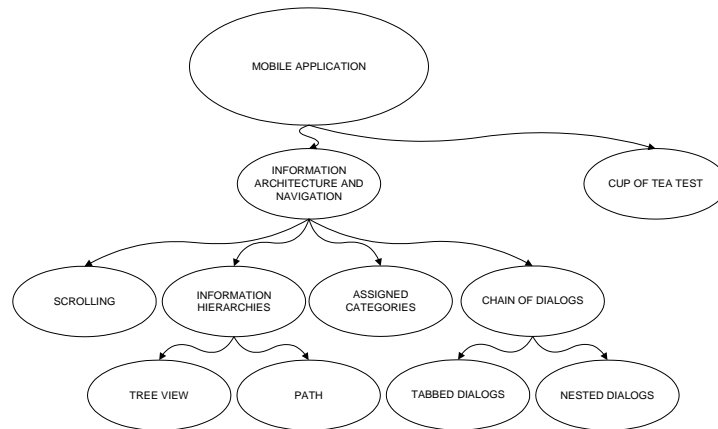


Abb. 9: Patterns zu Informations-Architektur und Navigation

Adaptivität und Adaptierbarkeit sind ebenfalls wichtige Gestaltungsprinzipien mobiler Anwendungen. Diese werden in den Patterns „User Customization“, „Context Awareness“ sowie „Task Adaptation“, auf welche im abstrakteren Pattern „Adaptability“ verwiesen wird (vgl. Abb. 4), ausführlicher erläutert.

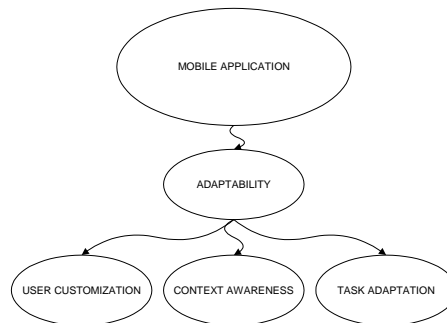


Abb. 4: Patterns zu Adaptivität und Adaptierbarkeit

2.1 Beispiel eines abstrakten Design Pattern: MOBILE APPLICATION

Kontext

Sie möchten eine Anwendung für ein mobiles Gerät entwickeln.

Problem

Mobile Geräte haben im Vergleich zu Desktop Computern spezielle Charakteristiken. Sie sehen anders aus und werden zu anderen Zwecken und unter anderen Bedingungen benutzt. Welche Möglichkeiten eröffnen sich durch diese Charakteristiken, und welche Einschränkungen resultieren aus den verschiedenen Eigenschaften?

Was muss bei der Gestaltung für mobile Geräte beachtet werden?

Problembeschreibung und Kräfte

Mobile Geräte sind tragbare Informations-, Organisations- und Kommunikationssysteme. Sie verteilen sich auf drei Kategorien: Mobiltelefone, Pager, PDAs und eine Kombination dieser Kategorien, der Smartphones.

Verglichen mit Desktop Computern haben sie spezielle Eigenschaften, welche einige Möglichkeiten, aber auch Einschränkungen mit sich bringen: sie werden zu anderen Zwecken genutzt; sie werden anders bedient; sie definieren Mobilität; sie besitzen eingeschränkte Eingabemechanismen; sie besitzen nur kleine Displays und limitierte Speichermöglichkeiten.

All diese verschiedenen Charakteristiken und Unterschiede erfordern spezielle Design-Ansätze für mobile Geräte und stellen einige Herausforderungen – und auch Möglichkeiten – für die Gestaltung dar.

Beispiele

Style Guides und Guidelines wie die Palm OS User Interface Guidelines [7], der Symbian UIQ Style Guide [8] oder die Nokia Series 60 UI StyleGuide [9] sind alle an die speziellen Charakteristiken und Anforderungen angepasst, die die entsprechenden Geräte mit sich bringen.

Die Benutzeroberflächen, die in diesen Guidelines beschrieben werden, sind alle für Geräte mit kleinen Bildschirmen, wenig Arbeitsspeicher und relativ einfachen Eingabemechanismen gestaltet. Sie alle versuchen, Anwendungen für diese Geräte so einfach benutzbar wie möglich zu machen.

Lösung

Wenn Sie Anwendungen für mobile Geräte gestalten, passen Sie das User Interface Design, die Informationsarchitektur und das Interaktionsdesign an die speziellen Charakteristiken dieser Geräte an.

Diagram

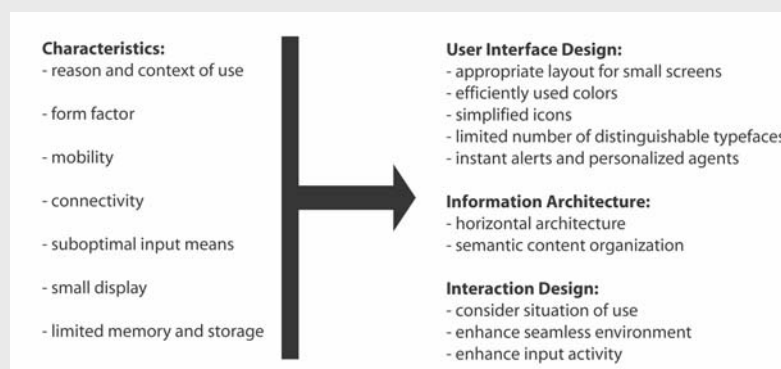


Abb. 5: Diagramm MOBILE APPLICATION

2.2 Beispiel eines konkreten Design Pattern: TREE VIEW

Kontext

Sie möchten hierarchische Informationen innerhalb Ihrer mobilen Anwendung organisieren und darstellen.

Problem

Wie können hierarchische Informationen dargestellt werden, ohne die Beziehungen der Daten zueinander zu verlieren und während eine einfache Navigation innerhalb dieser Daten sichergestellt werden soll.

Problembeschreibung und Kräfte

Wenn Anwender hierarchische Informationen durchsuchen und betrachten, so kann es passieren, dass sie die Orientierung innerhalb der komplexen Datenstruktur verlieren. Aus diesem Grunde ist es wichtig, den Kontext und die Beziehungen der Daten zueinander darzustellen.

Da mobile Anwendungen jedoch meist auf Geräten laufen, die nur eingeschränkte und kleine Displays besitzen, ist es schwierig diese komplexen Informationen und deren Beziehungen darzustellen und navigierbar zu machen.

Beispiele

Verschiedene mobile Anwendungen benutzen Baumdarstellungen, um komplexe hierarchische Informationen darzustellen. So zum Beispiel der Qtopia File Manager [10], der RESCO Explorer 2003 [11] und auch die Microsoft Pocket PC User Interface Guidelines [12] beschreiben Richtlinien zur Gestaltung von Baumstrukturen für Pocket PCs und Smartphones.

Lösung

Benutzen Sie Baumdarstellungen, um hierarchische Informationen darzustellen. Diese ermöglichen es, komplexe Informationen sowie die Beziehungen innerhalb der Daten zu visualisieren.

Sollen jedoch sehr tiefe Informationshierarchien dargestellt werden, so kann horizontales Scrolling notwendig werden. In solchen Fällen sollten andere Visualisierungstechniken wie beispielsweise Listen, die zusätzlich den Pfad zur Information darstellen, verwendet werden.

Diagramm

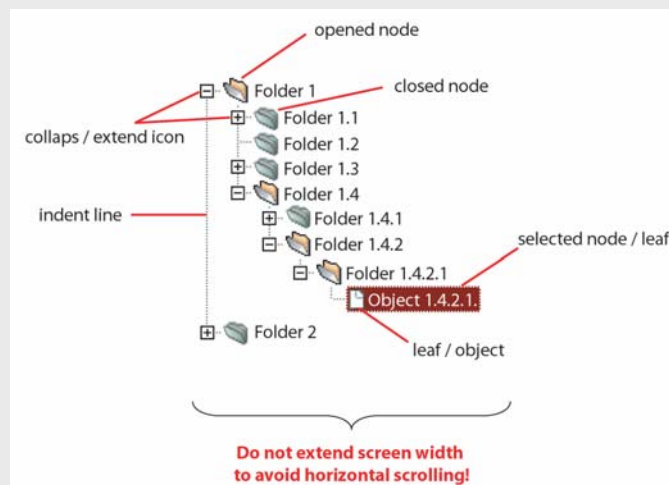


Abb. 6: Diagramm TREE VIEW

3. Design Patterns für Navigationssysteme auf mobilen Geräten

Ausgehend von der Wurzel der Pattern-Hierarchie – dem abstrakten Pattern „Navigation System“ – wird zunächst zwischen den verschiedenen Anwendungsbereichen von Navigationssystemen – der Fahrzeugnavigation und der Fußgängernavigation – unterschieden. Letztere unterteilt sich wiederum in Innen- und Außennavigation. Alle weiteren Patterns teilen sich dann nach den drei Hauptaufgaben von Navigationssystemen auf – der Routenplanung, der Wegführung sowie den Interaktiven Karten (vgl. Abb. 7).

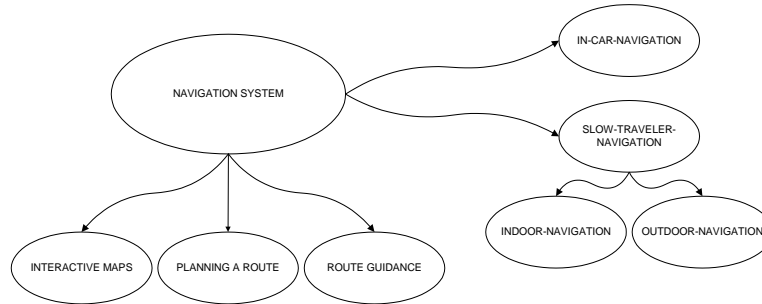


Abb. 7: Haupt-Pattern und Patterns zu den Anwendungsbereichen und Hauptaufgaben

Patterns der Routenplanung (vgl. Abb. 8) beschreiben unter anderem, wie ganze Routen als Favoriten abzuspeichern sind oder wie neue Routen mit Start, Ziel, Zwischenstationen und zu vermeidenden Bereichen angelegt werden können. Weiterhin wird erläutert, wie verschiedenen Eingabemöglichkeiten wie Texteingabe, Listenauswahl, Auswahl aus Adressbuch und Favoriten oder direkte Kartenwahl gestaltet werden können und wann welche dieser Möglichkeiten angeboten werden sollte.

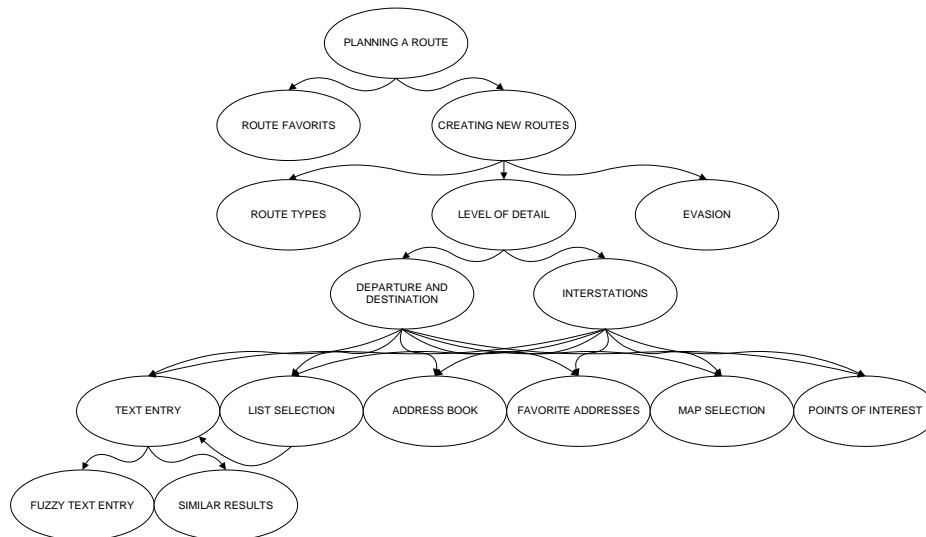


Abb. 8: Patterns zur Routenplanung

Zur Gestaltung interaktiver Karten werden verschiedene Patterns mit Möglichkeiten zum Zoomen und Verschieben (Panning) des Kartenausschnittes beschrieben, die je nach Rechenleistung des mobilen Gerätes und/oder der verwendeten Rendering-Algorithmen kontinuierlich oder in diskreten Schritten realisiert werden können.

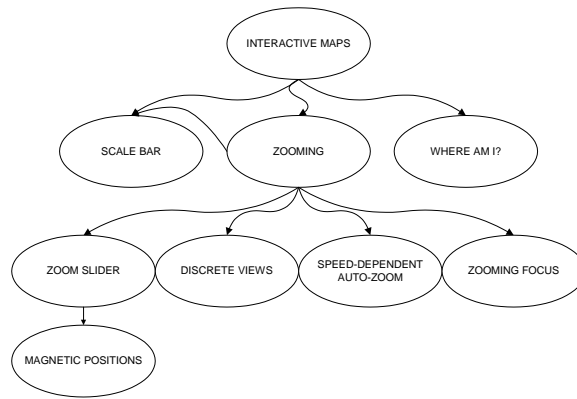


Abb. 9: Patterns zu Interaktiven Karten I

Weiterhin sind Interaktionsmechanismen wie eine Interaktive Legende, eine Helikopter-Funktion zum kurzzeitigen Herauszoomen aus der Karte oder ein navigierbares Übersichtsfenster beschrieben, welche dem Anwender Orientierung in der Karte geben (vgl. Abb. 9 und 10).

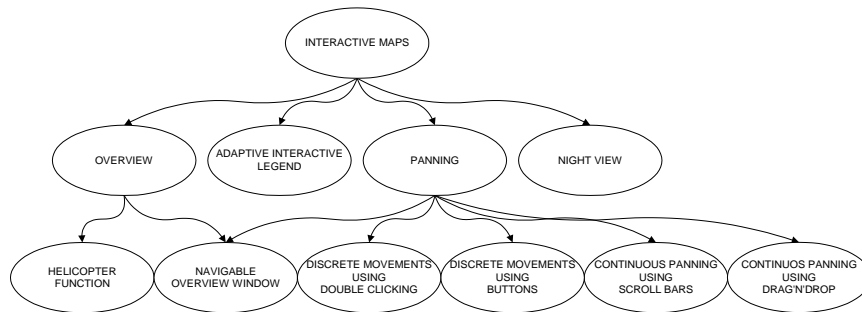


Abb. 10: Patterns zu Interaktiven Karten II

In den Wegführungs-Patterns (vgl. Abb. 11) werden Lösungen zur Darstellung einer konkreten Route und der Navigation entlang dieser gegeben. So können Routen einerseits als Ganzes dargestellt werden, beispielsweise in einer interaktiven Karte oder als Anweisungsliste. Eine inkrementelle Darstellung der Abbiegeanweisungen in 2D, 3D oder einfach als textuelle Ausgabe auf dem Bildschirm oder als Sprachausgabe kann jedoch auch sinnvoll sein. Weiterhin werden Patterns zu konkreten Elementen einer Anweisung wie Entfernungsbalken, Pfeildarstellungen in Piktogrammform oder auch zu wichtigen Landmarks beschrieben.

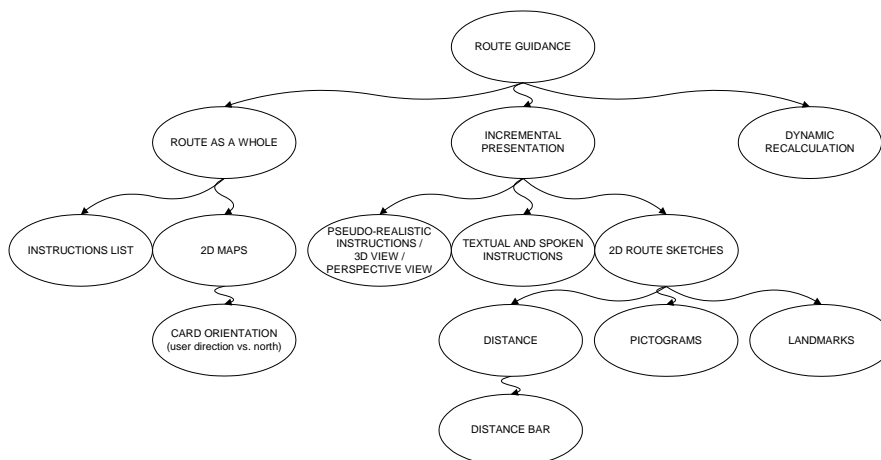


Abb. 11: Patterns zur Wegführung

3.1 Beispiel eines abstrakten Design Pattern: NAVIGATION SYSTEM

Kontext

Sie wollen eine Anwendung entwickeln, die Menschen helfen soll, von einem Ort zu einem anderen zu gelangen.

Problem

Wie können Personen am besten von einem Ort zu einem anderen geführt werden? Welche Funktionen sollte ein entsprechendes Hilfsmittel bereitstellen, damit diese Personen bei ihrer Aufgabe am besten unterstützt werden?

Problembeschreibung und Kräfte

Menschen sind immer unterwegs. Sie benutzen hierbei die unterschiedlichsten Fortbewegungsmittel, wissen aber selten, wohin genau sie hin müssen und wie sie dort hin gelangen können.

Aus diesem Grunde existieren schon seit langem Karten für alle Gegenden der Erde. Diese Karten, welche zumeist aus Papier bestehen, haben jedoch einige entscheidende Nachteile: sie sind nicht erweiterbar; sie bieten keine Möglichkeit zur automatischen Navigation; es ist meist schwierig, exakte Positionen zu finden und die Anwender müssen sich eine gewählte Strecke merken oder sie müssen die Karte immer wieder betrachten.

Beispiele

Das wahrscheinlich erste Projekt, welches sich mit der Gestaltung von Benutzeroberflächen für Navigationssysteme beschäftigte war UI+IV (User Interface and Information Visualization), welches 1989 – 1992 von Motorola durchgeführt wurde [6]. Ziel des Projektes war die Entwicklung eines Fahrzeugnavigationssystems, welches unter Beachtung von Usability-Prinzipien so gestaltet werden sollte, dass der Anwender nicht zu sehr vom Verkehr abgelenkt wird.

Heutzutage gibt es die verschiedensten Navigationssysteme für alle möglichen Anwendungsgebiete und Geräte. Zu ihnen zählen Fahrzeugnavigationssysteme, welche mittlerweile fast serienmäßig in neuen Autos integriert werden, sowie Systeme für PDAs. Zusätzlich existieren Fußgängernavigationssysteme, welche Menschen durch Städte, Museen oder über Messen führen. Sogar für Mobiltelefone gibt es einige Lösungen.

Lösung

Entwickeln Sie ein interaktives Navigationssystem, mit dessen Hilfe Menschen einfach Karten anschauen, Routen planen und entlang zuvor geplanter Strecken geführt werden können.

Unterstützen Sie die Anwender eines solchen Navigationssystems bei der Eingabe von Daten wie Start- und Zieladressen, stellen Sie nur die notwendigen Informationen dar und ermöglichen Sie eine einfache Wegführung, welche den Anwender von seiner aktuellen Position (oder jeder anderen Gewünschten) zum gewünschten Ziel leitet.

Diagramm

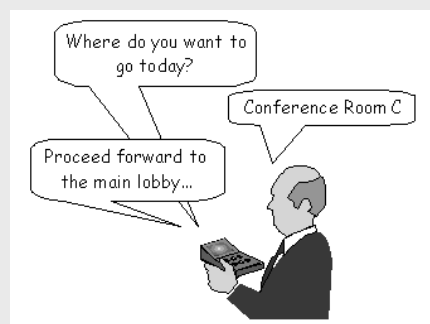


Abb. 12: Diagramm NAVIGATION SYSTEM

3.2 Beispiel eines konkreten Design Pattern: HELICOPTER FUNCTION

Kontext

Sie wollen eine Überblicks-Funktion innerhalb interaktiver Karten gestalten. Die Funktion soll einen kurzen, aber informativen Überblick über die Umgebung des aktuellen Kartenausschnittes geben.

Problem

Wie muss eine Funktion gestaltet werden, die auf Knopfdruck einen direkten und informativen Überblick einer Karte bietet? Welche Anforderungen hat eine solche Funktion?

Problembeschreibung und Kräfte

Anwender, die sich mit einer hohen Zoom-Stufe innerhalb einer interaktiven Karte bewegen, verlieren leicht die Orientierung, wo der gerade dargestellte Kartenausschnitt innerhalb der Gesamtkarte liegt. Es besteht also der Bedarf nach einer Überblicks-Funktion. Diese wird spontan benötigt und sollte deshalb einfach aufrufbar sein und eine schnelle Orientierung bieten.

Beispiele

In Falk City Guide [13] existiert eine Helikopter-Funktion, die, solange der Anwender das Symbol der Funktion anklickt, aus der interaktiven Karte herauszoomt. Anschließend verweilt die Karte für wenige Sekunden in diesem Zustand und zoomt danach auf den vorherigen Ausschnitt zurück.

Der Online-Routenplaner Map24 [14] bietet ebenfalls eine solche Funktion – hier allerdings Rocket-Funktion genannt. Obwohl die Funktionalität dieselbe ist, trifft die Raketen-Metapher weniger zu, da Raketen im Gegensatz zu Helikoptern meistens nicht mehr zur Erde zurückkehren.

Lösung

Bieten Sie eine Helikopter-Funktion, die nach dem Aufruf aus dem aktuellen Kartenausschnitt herauszoomt, in dieser Übersicht einen Moment verweilt und anschließend automatisch in die vorige Ansicht zurückzoomt. Die Funktion sollte einfach auf Knopfdruck aufrufbar sein und lange genug im Überblicks-Zustand verweilen, so dass sich der Anwender Orientieren kann.

Diagramm

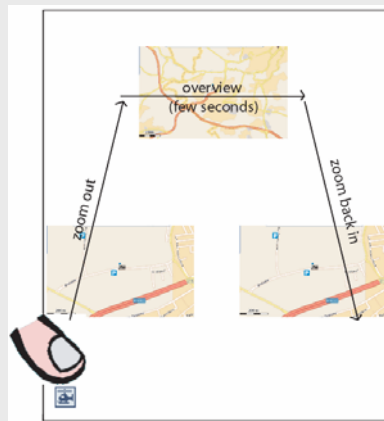


Abb. 13: Diagramm HELICOPTER FUNCTION

4. Einsatz der Patterns in Entwicklungsprojekten

Die hier vorgestellten Design Patterns entstanden im Projekt, um mit relativ wenig Aufwand wesentliche Gestaltungsprinzipien zu formulieren und einfach im Team zu kommunizieren. Hierin zeigt sich die Stärke des Design Pattern-Ansatzes: auf allgemeiner Ebene können wichtige Richtlinien dargestellt und verständlich gemacht werden. Darüber hinaus eignen sich Design Pattern-Sprachen zur Sammlung und integrierten Darstellung von Ergebnissen aus Usability-Studien und -Projekten bezogen auf bestimmte Anwendungsbereiche. Das Wissen über gewählte Design-Patterns und ihre ergonomische Qualität lässt sich auf diese Weise gut aufbereiten und vermitteln. Langfristig können so erweiterbare Bibliotheken bewährter Patterns entstehen, auch weil Pattern-Sprachen durch ihren modularen Aufbau leicht gepflegt und erweitert werden können.

Design Patterns bieten so ein leicht handhabbares Format, das die Sammlung abstrakter Prinzipien und Gestaltungswissen unterstützt. Sie ergänzen sich damit gut zu UI-Spezifikationen oder Styleguides, sind jedoch kein vollständiger Ersatz dafür. Insbesondere bleiben sie in der konkreten grafischen Ausgestaltung des UI unkonkret und müssen deshalb um solche Beschreibungen ergänzt werden.

5. Literaturverzeichnis

- [1] Alexander C., Ishikawa S., Silverstein M., Jacobson M., Fiksdahl-King I. and Angel S., *A Pattern Language*. Oxford University Press (1977).
- [2] Alexander C., *The Timeless Way of Building*. Oxford University Press (1979)
- [3] Gamma E., Helm R., Johnson R. and Vlissides J., *Design Patterns: Elements of Reusable Object Oriented Software*. Addison-Wesley (1995)
- [4] Borchers J., *A Pattern Approach to Interaction Design*. John Wiley & Sons (2001)
- [5] Noble J. and Weir C., *A window in your pocket: Some small patterns for user interfaces*. In Proc. European Pattern Languages of Programs. The Hillside Group, Inc. (2001)
- [6] Aaron, M. (2000): *Designing the User Interface for a Vehicle Navigation System: A Case Study*. In Bergman, E., editor, *Information Appliances and Beyond: Interaction Design for Consumer Products*, Morgan Kaufmann, San Francisco, pp. 205-255.
- [7] Ostrem, J. (2002): *Palm Os User Interface Guidelines*, Document Number 3101-001, http://www.palmos.com/dev/support/docs/ui/uiuide_front.html
- [8] Symbian Ltd. (2002): *Symbian UIQ style guide*, http://www.ericsson.com/mobilityworld/sub/open/technologies/epoc/docs/uiq_style_guide
- [9] Nokia Corporation (2002): *Nokia Series 60 UI StyleGuide v4.0*, <http://www.forum.nokia.com/main/>
- [10] Trolltech Inc.; <http://www.trolltech.com/products/qtopia/index.html>
- [11] Resco, Ltd.; <http://www.resco-net.com/explorer.asp>
- [12] Microsoft Corporation (July 2003): *Pocket PC User Interface Guidelines* http://msdn.microsoft.com/library/en-us/ui_guide_ppc/hm/Tree_View_Controls.asp
- [13] Falk City Guide; <http://www.falk.de>
- [14] Map24 - Das Internetportal für interaktive Karten; <http://www.map24.de> in May 2003

6. Der Autor

Rainer Gibbert verfasste seine Diplomarbeit im Competence Center Human-Computer Interaction des Fraunhofer Institut für Arbeitswirtschaft und Organisation in Stuttgart. Im Rahmen des Projektes SAiMotion, welches er als wissenschaftliche Hilfskraft unterstützte, erarbeitete er als Diplomarbeit eine Design Pattern-Sprache für mobile Applikationen mit dem Schwerpunkt Navigationssysteme, die Grundlage dieses Beitrags ist.

Rainer Gibbert studierte im Studiengang „Digitale Medien“ an der Fachhochschule Kaiserslautern, Standort Zweibrücken, und erhielt dort im Oktober 2003 sein Diplom in Informatik.