

ADF Mobile Patterns: Standing on the Shoulders of Giants

Frank Nimphius
Oracle

keywords

ADF Mobile, Android, iOS, Pattern, Best Practices

Einleitung

Pattern sind Handläufe in der Anwendungsentwicklung die Software Entwickler zu gutem Design führen sollen. Im Laufe der letzten Zeit haben sich Handläufe von universellen, generischen Mustern (Blueprints) hin zu produktspezifischen Empfehlungen gewandelt (Best Practices).

Der Grund hierfür liegt meiner Meinung nach darin, daß Entwicklungszyklen und Innovationszyklen immer kürzer werden. Anwendungsentwickler müssen immer schneller auf Veränderungen reagieren und dabei ein immer größeres Potfolio an Techniken abdecken. Da bleibt wenig Zeit für Fehler und dem Entdecken von Best Practices und Design Patterns.

Als Folge sind zunehmend moderne produktspezifische Handläufe (ich werde sie ab jetzt als Pattern bezeichnen) gefragt die in der Regel von User Interface Professionals (UX) oder dem Produkthanbieter basierend auf allgemeinen Mustern und den Erfahrungen in der Zusammenarbeit mit Kunden beruhen.

Dabei lassen sich für die Entwicklung von ADF Mobile Anwendungen allgemeine Mobile Pattern sowie ADF Mobile produktspezifische Muster nutzen, die dann auch noch in Architektur-, Layout-, Funktionale- und Navigation-Pattern eingeteilt werden können.

Um Mißverständnissen vorzubeugen: Natürlich gelten klassische Pattern wie die der sogenannten Gang of Four (GoF) auch in der Entwicklung von Anwendungen für mobile Endgeräte. Allerdings nehmen Frameworks wie ADF Mobile dem Entwickler einige Arbeit in der Implementierung ab, so daß die Entwicklung nicht ausschließlich klassisch objekt-orientiert erfolgt.

Das Geheimnis des Erfolges

Gutes Design überlässt nichts dem Zufall. Usability Research und Design (UX) ist eine weitestgehend nicht öffentliche Arbeit (Oracle's UX Team mag da die Ausnahme sein) innerhalb von produzierenden Unternehmen, die Aspekte moderner Psychologie und Anwenderstudien verwendet um Produkte auf Benutzbarkeit hin zu optimieren.

So spielen Navigationsstrukturen, Farb- und Helligkeits-Kontraste, Leserichtung, und Schriften bei browser basierten Software-Anwendungen eine übergeordnete Rolle, während Haptik, die taktile Wahrnehmung von Oberflächen, zum Beispiel keinen großen Einfluß auf die Akzeptanz hat.

Das ändert sich allerdings wenn die Anwendung auf einem mobile Endgerät ausgeführt wird und sich das Gerät in den Händen des Anwenders befindet. Das mobile Endgerät – Telefon oder Tablet - ist nun Teil des Bedienungskonzeptes. Die Anwendung diktiert wann der Anwender- oder die Anwenderin das Display überstreicht, wann ein Finger auf das Display gepresst werden muß, wann zwei Finger benutzt werden und wann das Gerät besser gedreht werden sollte.

Plötzlich bekommt die unterschiedliche Haptik mobiler Endgeräte (Smart Phone, Tablet, Mini-Tablet) Einfluß auf die Anwender-Akzeptanz einer Software Anwendung.

Das Geheimnis des Erfolges beruht demnach nicht ausschließlich auf der generellen Verfügbarkeit einer Anwendung auf mobilen Endgeräten. Stattdessen wird der Erfolg einer Anwendung, und somit Ihres mobilen Auftritts, durch das "Bewußtsein" und die Anpassungsfähigkeit der Anwendung auf das jeweilige mobile Endgerät des Nutzers erreicht.

Mobile Layout Pattern

Mobile Benutzeroberflächen sind durch die reduzierte Größe des Displays begrenzt, was wiederum Auswirkung auf die Menge darstellbarer Information pro Seite sowie das Layout hat.

Für Sie als Anwendungsentwickler bedeutet das, daß Sie eine klare Bildsprache in der Form von Icons den Prompts oder Labeln vorziehen. Ganz generell gilt, daß weniger mehr ist und eine Konzentration auf das Wesentliche erfolgen muß.

Zu oft läd uns die Deutsche Sprache zu längeren Wort und Satzungestümen ein: "Wenn Sie an diese Stelle angelangt sind, betätigen Sie bitte die Bremse Ihres Fahrzeuges bis es vollständig steht. Warten Sie 2 Sekunden und vergewissern Sie sich, daß die Kreuzung frei ist bevor Sie Ihre Fahrt fortsetzen".

Auf dem mobilen Endgerät würde dieser Hinweis dann doch besser durch ein Stop Symbol angezeigt. Achten Sie bei der Auswahl von Bildern und Icons auf eine klare Bildsprache und Erkennbarkeit bei wechselnden Lichtbegebenheiten (Ja, Benutzer mobiler Endgeräte tendieren dazu sich zu bewegen).

Vorsicht ist geboten bei allen Formen von innovativen Design, da Anwender mobiler Endgeräte eine gewisse Implementierungs-Normalität erwarten um intuitiv mit einer Applikation zu arbeiten. Online Hilfen zum Verstehen der Benutzeroberfläche sind äußerst unüblich und drücken eher die Einsicht des Entwicklers aus, daß das mit dem mobile Design noch nicht so richtig geklappt hat. Sie werden daher um bestimmte allgemein akzeptierte Layout-Pattern nicht herum kommen:

Springboard – Ein Zeilen- und Spaltenbasiertes Layout zur Darstellung von Anwendungs Funktionen über Icons. Das Springboard ist ein typisches "landing-pad" multifunktionaler Anwnedungen.

Um Springboard kreativ zu gestalten, zum Beispiel um bestimmte Funktionen hervorzuheben, bietet es sich an bestimmte Icons über zwei Spalten oder zwei Zeilen hinweg anzuzeigen, oder in einer anderen Farbe darzustellen. Das Microsoft Windows 8 Kacheldesign kann Ihnen dabei als kreative Inspiration dienen.

Allerdings empfehle ich von der Idee Abstand nehmen Kacheln horizontal auszuweiten, so daß der Anwender horizontal scrollen muß. In Sachen Springboard-Navigation hat sich das vertikale Scrollen oder die Verwendung von "Second-Level" Springboards durchgesetzt.

Menu-Toolbar – Eine Navigationsleiste am Kopf- oder Fußende einer Anwendung, die optional ausgeblendet werden kann. Die Navigationsleiste ermöglicht innerhalb der Anwendung zwischen verschiedenen Funktionalitäten zu navigieren.

Listen –Listen sind ein Standard Pattern wenn es darum geht Master-Detail Verhalten darzustellen. Dabei scrollt der Anwender die vornehmlich vertikale Liste um mittels Auswahl eines Listenelements auf eine Folgeseite mit den Detail Informationen zu gelangen. Listen sind ein Pattern das wohl in jeder Anwendung zu finden ist.

Reduzierte Freitext-Eingaben - Mobile Endgeräte haben ein Manko: die virtuelle Tastatur. Ich weiß nicht wie es Ihnen geht, mir fällt es jedenfalls schwer längere Texteingaben fehlerfrei einzutippen und fühle mich manchmal wie bei "Verstehen Sie Spaß". Um diesem Manko entgegenzuwirken verwenden

die meisten mobilen Anwendungen vordefinierte Listen (Radio Gruppen, Checkboxes, Auswahllisten) und Auswahlelemente (Kalender, Slider).

Label-on-top – Es ist üblich Label über den Eingabefeldern zu positionieren statt links oder rechts daneben. Der Grund hat wieder mit dem Scroll-Verhalten zu tun. Eine Mischung aus horizontalem Scrolling (verursacht durch lange Label) und vertikalem Scrolling (verursacht durch Eingabemasken die nicht als Ganzes in die verfügbare Länge des Displays passen) ist einfach nicht ergonomisch und als umständlich zu betrachten.

Ich belasse es bei den oben erwähnten Beispielen, da ich es nicht schaffen werde auf 6 Seiten alle möglichen im Einsatz befindlichen Pattern zu beschreiben.

Funktionale Pattern

Funktionale Pattern beschreiben Implementierungsvorschläge. So ist es möglich in ADF Mobile spezielle Features für eine bestimmte Display Größe (Smart Phone vs. Mini-Tablet vs Tablet) zu entwickeln und dynamisch ein-, bzw. auszublenden.

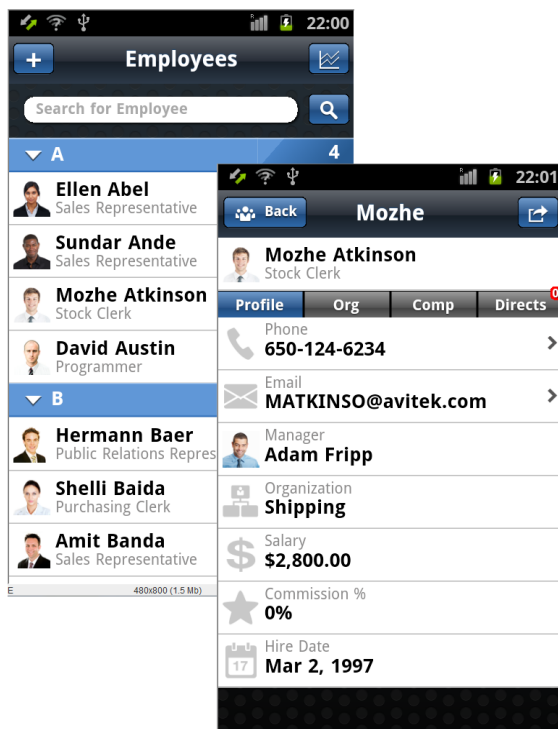


Abb. 1: ADF Mobile HR Sample Feature auf dem Smart Phone

Feature Constraints ist eine Alternative zu den sonst verwendeten adaptiven Layouts die durch CSS kontrolliert werden. Abbildung 1 und Abbildung 2 zeigen die Verwendung von Feature Constraints in der HR Beispiel- Anwendung, die mit ADF Mobile ausgeliefert wird.

Abbildung 1 zeigt das Employee Feature für Smart Phones. In diesem Feature werden zwei Seiten für die Darstellung der Employee Liste und der Employee Details verwendet.

Abbildung 2 zeigt das Feature, das für Tablets entwickelt wurde. Hier erfolgt die Employee-Asuwahl Employees über eine Liste innerhalb eines Split Panels. Detail-Informationen werden auf der selben

Seite – nebst geografischer Karte – angezeigt. Die Unterscheidung zwischen den unterschiedlichen Geräten erfolgt deklarativ über sogenannte Constraints.

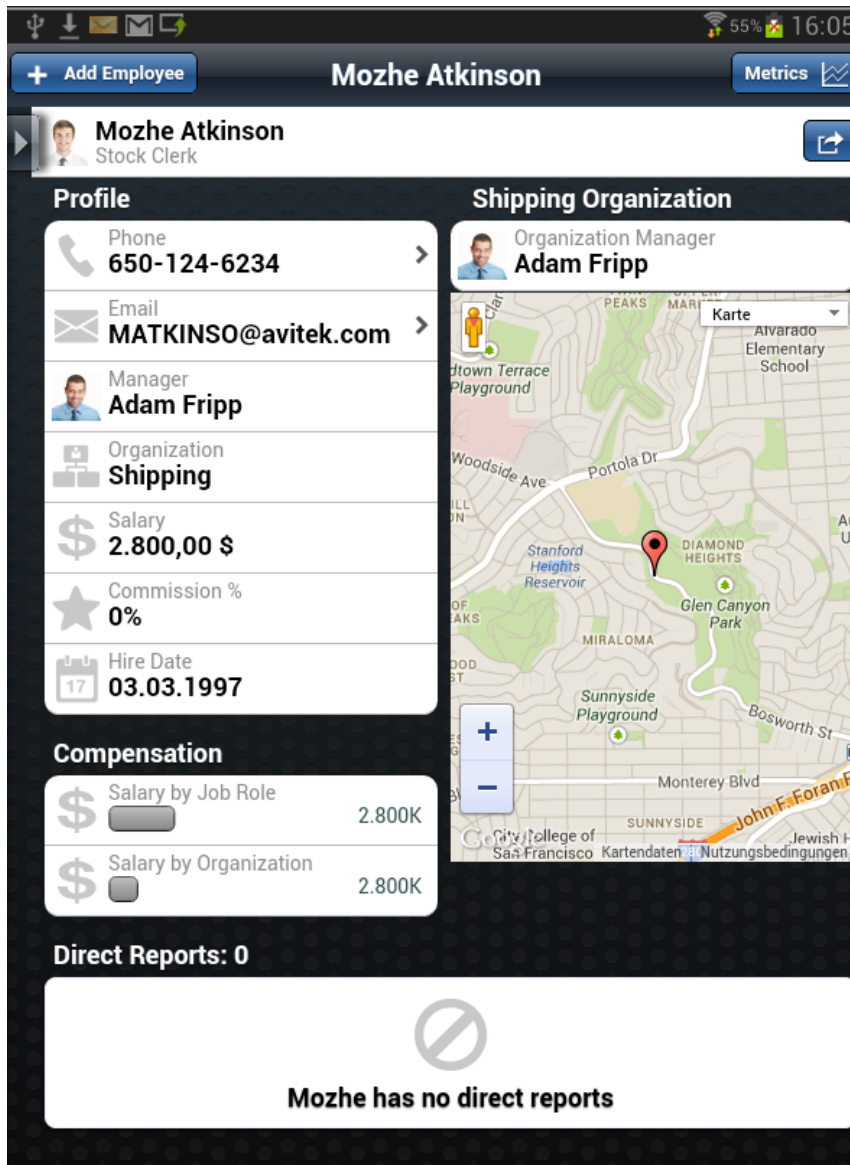


Abb. 2: ADF Mobile HR Sample Feature auf dem Tablet

Andere ADF Mobile Pattern in dieser Kategorie sind Task Flow, Form-Cancel, Partial Content Refresh, Swipe Data Browsing, etc.

Navigation Pattern

Halten Sie Navigationsstrukturen einfach und vermeiden Sie tiefe Schachtelungen. In ADF Mobile können Navigationpfade auf verschiedene Art und Weise implementiert werden:

Ungebundene Seiten Navigation – Navigation zwischen verschiedenen Mobile-Seiten innerhalb eines bounded oder unbounded Task Flow.

Task Flow Navigation – Die Navigation zwischen zwei Task Flows – kann man auch als Sub-Flow Navigation bezeichnen.

Feature Navigation – Navigation zwischen zwei Features, wobei Features unterschiedlichen Technologien angehören können (AMX, HTML5, Remote URL).

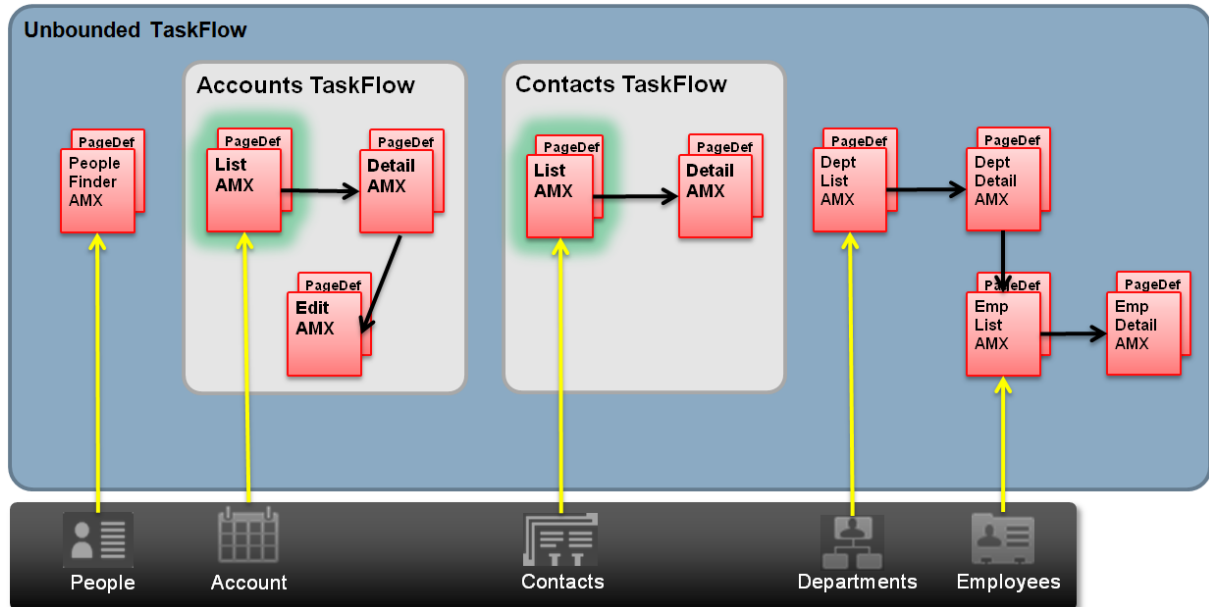


Abb. 3: Mobile Navigation Pattern

Abbildung 3 zeigt verschiedenen ADF Mobile Navigations-Pfade in einer Übersicht. Über die Feature Navigation lassen sich Hub & Spoke (Wagenrad) Pattern einfach implementieren. Innerhalb der Spoke (Speiche) erfolgen dann Seiten- oder Task Flow Navigationen.

Tiefe in der Navigation – und wiederverwendbarkeit innerhalb einer Anwendung – erzielen Sie durch Task Flow Navigation.

Daten-bezogene Pattern

ADF Mobile – und mobile Anwendungen im Allgemeinen – haben spezielle Anforderungen an die Menge und Qualität der übertragenen Daten.

Internet Verbindungen sind für mobile Anwendungen nicht immer stabil, so daß Daten lokal vorgehalten werden müssen. Darüber hinaus gilt es die Band-breite der Internetverbindung zu berücksichtigen wenn es um das Volumen der Daten geht.

In wesentlichen geht es darum eine lokale Caching-Strategie zu implementieren (und zwar bis in die server-seitige Datenbank hinein, um zu vermeiden, daß lokal gehaltene Information zwischenzeitlich geänderte Datensätze in der Datenbank überschreibt. Dazu lassen sich vier Pattern identifizieren:

- Online Lesen und Schreiben
- Offline Lesen und online Schreiben
- Offline Lesen und Schreiben

- Volle Synchronisation

In ADF Mobile können lokale Caching Strategien sowohl die lokale SQLite Datenbank, als aber auch lokale POJOs verwenden.

Darüber hinaus ist die Datenmenge zu kontrollieren und zwar vertikal (die Menge an Datensätzen) und horizontal (Die Menge an Attributen pro Datensatz), sowie die Übertragungstechnologie (SOAP oder REST). Oftmals entscheidet der Anwendungsfall (Usecase) welche implementierung erfolgt.

Und zum Schluß noch ein Hinweis auf die mögliche Anforderung Daten sozusagen im Transit zu bearbeiten (Kalkulation, Filter, Type-Casting). Auch hier gibt es verschiedene Ansätze die sich in Pattern gliedern lassen.

Fazit

Moderne Pattern sind längst nicht mehr unbedingt generischer Natur und oftmals Implementations-Spezifisch um dem Entwickler eine helfende Hand zu reichen (Handläufe).

Viele der für mobile Anwendungen dokumentierte Pattern finden sich in ADF Mobile Komponenten, oder Framework Funktionen wieder. Sie brauchen Sie also nicht speziell zu programmieren.

Andere Pattern können als Composites aus den bestehenden UI und Funktionalen Komponenten, sowie Java und JavaScript, implementiert werden. Hier lohnt es sich einmal etwas inne zu halten und sich mit den allgemein üblichen mobile Pattern zu Beschäftigen, ein Wireframe für die eigene Mobile Anwendung zu erstellen und erst dann zu programmieren. Es zahlt sich aus!

Kontakt :

Frank Nimphius

Oracle

E-Mail frank.nimphius@oracle.com

Internet: www.oracle.com