



Konzeption und Implementierung einer touchgesteuerten Oberfläche für einen konfiguratorbasierten Produktkatalog

Bachelorthesis

für die Prüfung zum

Bachelor of Science

des Studienganges Angewandte Informatik

an der Dualen Hochschule Baden-Württemberg Karlsruhe

von

Dane Leube

27.03.2013

Bearbeitungszeitraum

Matrikelnummer, Kurs

Ausbildungsfirma

Betreuer

Gutachter

12 Wochen

1313394, TAI10B2

CAS Software AG, Karlsruhe

Dr. Michael Klein

Dipl. Inform. Thorsten Schlachter

Erklärung

gemäß § 5 (2) der „Studien- und Prüfungsordnung DHBW Technik“ vom 18. Mai 2009.

Ich habe die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Karlsruhe, 27.03.2013

Dane Leube

Zusammenfassung

Ein Abstract ist eine prägnante Inhaltsangabe, ein Abriss ohne Interpretation und Wertung einer wissenschaftlichen Arbeit. In DIN 1426 wird das (oder auch der) Abstract als Kurzreferat zur Inhaltsangabe beschrieben.

Objektivität soll sich jeder persönlichen Wertung enthalten

Kürze soll so kurz wie möglich sein

Genauigkeit soll genau die Inhalte und die Meinung der Originalarbeit wiedergeben

Üblicherweise müssen wissenschaftliche Artikel einen Abstract enthalten, typischerweise von 100-150 Wörtern, ohne Bilder und Literaturzitate und in einem Absatz.

Quelle <http://de.wikipedia.org/wiki/Abstract> Abgerufen 07.07.2011

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Ziel der Arbeit	2
1.3	Vorgehensweise	2
2	Grundlagen	4
2.1	Produktkonfiguratoren	4
2.1.1	CAS Configurator Merlin Enterprise	5
2.1.2	Konfigurator des Kunden Airbus	6
2.2	Grundlegende Anforderungen	7
2.2.1	Abgrenzung der Anforderungen	7
2.2.2	Nicht-Funktionale Anforderungen	9
2.2.3	Funktionale Anforderungen	10
3	Konzeption der Anwendung	11
3.1	Aktueller Konfigurationsprozess	11
3.2	Workflow Modellierung	12
3.2.1	Mobiler Konfigurationsprozess	13
3.2.2	Workflow der App	13
3.3	Auswahl der Anwendungsplattform	15
3.3.1	Native Anwendungen	16
3.3.2	Web Anwendungen	17
3.3.3	Hybride Anwendungen	17
3.3.4	Abwägung	17
4	Entwurf der Benutzeroberfläche	18
4.1	Windows 8	18
4.1.1	Bedienkonzepte	18
4.1.2	MVVM	18
4.2	Design der Ansichten	18
4.3	Interaktion der Ansichten	18

5	Implementierung	19
5.1	Implementierung des Katalog-Workflows	19
5.2	Implementierung des Konfigurations-Workflows	19
6	Evaluation der Anwendung	20
6.1	Zielkriterien	20
6.2	Testergebnisse	20
7	Fazit und Ausblick	21
	Abbildungsverzeichnis	i
	Literaturverzeichnis	iii

1 Einleitung

1.1 Motivation

Auf dem heutigen Käufermarkt wird der Verkauf von individuellen Produkten immer wichtiger. Da Produkte sich immer weniger in ihren Hauptfunktionen unterscheiden, muss dem Kunden die Möglichkeit gegeben werden sein Produkt individuell zusammenzustellen. Dieser Trend wird als *Mass Customization* [MASS] bezeichnet. Die Konfigurationsmöglichkeit des Produktes verursacht zusätzlichen Aufwand, da während der Konfiguration die technische Realisierbarkeit geprüft werden muss. Produktkataloge, die Informationen über die Komponenten eines Produktes beinhalten, sind meist in Papierform vorhanden. Aus diesem Grund wird die Prüfung, ob das Produkt realisiert werden kann normalerweise von einem erfahrenen Mitarbeiter händisch durchgeführt.

Für die Unterstützung dieses Prozesses werden immer mehr computergestützte Systeme eingesetzt. Diese verwenden modellierte Regeln, die eine vorhandene Produktwelt abbilden. Für die Anzeige, bzw. das Zusammenstellen des Produktes werden vermehrt mobile Endgeräte verwendet. Durch ihre hohe Verfügbarkeit, sowie einfache Handhabung bieten sie entscheidende Vorteile beim Verkaufsgespräch mit dem Kunden. Ein Fokus bei der Entwicklung einer Anwendung für ein mobiles Endgerät (im folgenden App genannt) ist eine intuitive Bedienung durch Touch-Gesten. Ein passendes Bedienkonzept hilft beim Durchführen einer Zusammenstellung für den Kunden (Konfiguration). Die Herausforderung besteht im Zusammenführen der komplexen Konfigurationslogik mit einem passenden intuitiven Bedienkonzept.

1.2 Ziel der Arbeit

Im Rahmen dieser Arbeit soll eine prototypische Tablet-PC Anwendung erstellt werden. Der Inhalt dieser App besteht zum Einen aus dem vorhandenen Produktkatalog in Papierform und zum Anderen aus einer Anbindung an den Produktkonfigurator. Die Anwendung muss ein Feedback geben können, ob die derzeitige Auswahl technisch umsetzbar ist. Hierzu soll der vorhandene Konfigurator eines Kunden angebunden werden.

Der Fokus der Anwendung liegt in der Konzeption eines Bedienkonzeptes, welches die vorhandenen Möglichkeiten der Technologie ausnutzt. Der Nutzen für den Kunden soll eine Vereinfachung des derzeitigen Configurations-Prozesses mithilfe der Software sein. Ebenfalls sollen durch den mobilen Einsatz der App neue Möglichkeiten bei der Konfiguration entstehen. Der Papierkatalog soll hierfür mit den technologischen Möglichkeiten erweitert und dadurch vereinfacht werden. Für die Zielerreichung ist die Auswahl der Plattform, sowie die Entscheidung, ob Native- oder Web-Anwendung entscheidend.

1.3 Vorgehensweise

Am Anfang der Arbeit wird der Workflow des Kunden nachvollzogen. Dieser beinhaltet die Schritte von der Auswahl eines Produktes im Katalog, über die Prüfung der Machbarkeit, bis zur Bestellung des gewählten Elements. Ist der Prozess verstanden, kann darauf aufbauend die Entscheidung der Plattform folgen. Hierbei werden die drei vorhandenen Möglichkeiten mobil, hybrid und nativ untersucht. Ist die Art der Anwendung ausgewählt, kann mit der Entscheidung der Technologie fortgefahren werden. Nach der Auswahl wird die Anwendung konzipiert, wobei die intuitiven Bedienelemente der gewählten Technologie verwendet werden. Nach der Konzeption erfolgt die Implementierung des Prototyps. Anschließend wird eine Evaluation erfolgen, welche die Anwendung nach den zuvor bestimmten Kriterien bewerten soll.

Kapitel 2 behandelt die grundlegenden Anforderungen der Anwendung und gibt einen Überblick des Produktes Merlin Enterprise, sowie dessen Einsatz beim Kunden. In Kapitel 3 wird die Konzeption der Anwendung und die darauf resultierende Entscheidung über die passende Plattform behandelt. Anschließend erfolgt in Kapitel 4

das Entwerfen der einzelnen Ansichten, bevor in den Kapiteln 5 und 6 die Implementierung und Evaluation der Anwendung beschrieben wird. Im letzten Abschnitt wird es einen Ausblick und ein Fazit über die gesamte Arbeit geben.

2 Grundlagen

In diesem Kapitel wird zuerst auf das Umfeld der Arbeit im Bereich der Produktkonfiguratoren eingegangen. Das Produkt CAS Merlin Enterprise wird anschließend vorgestellt, sowie auf die kundenspezifischen Konfigurationsprozesse eingegangen. Zum Abschluss werden die Anforderungen der Anwendung dargestellt.

2.1 Produktkonfiguratoren

Das Ziel des Produktkonfigurators ist es, produktspezifisches Wissen für die Anwender bereit zu stellen. Dieses hilft bei der individuellen Zusammenstellung des Produktes. Ein solches System wird in die Kategorie der Expertensysteme[PUPPE] oder wissensbasierte Systeme eingeordnet. Der Aufbau eines solchen Systems ist in Abbildung 2.1 zu sehen.

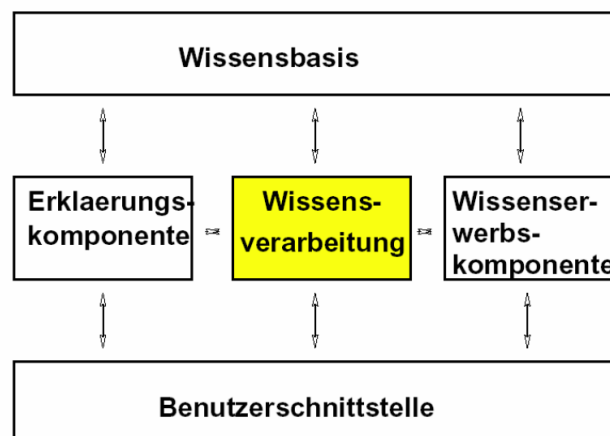


Abbildung 2.1: Aufbau eines Expertensystems [KELLER, s.6]

Die zentrale Komponente ist die *Wissensverarbeitung*. Diese hat auf alle weiteren Komponenten Zugriff und interagiert mit diesen. Es werden die erhaltenen Fakten mithilfe

der vorhandenen Regeln verknüpft. Aus der Verknüpfung werden neue Fakten gewonnen, die auf der *Benutzerschnittstelle* angezeigt werden. Die Wissensbasis ist für das Speichern des Expertenwissens in Fakten und Regeln zuständig. Die Speicherung der Daten kann auf folgende zwei Arten geschehen[EXPERT]:

- **generisch:** unabhängig vom aktuellen Anwendungsfall. Meist in einfachen Wenn-Dann-Regeln oder auf einem Modell beruhend.
- **fallspezifisch:** zur Lösung eines konkreten Anwendungsfall

Die Pflege dieser Basis erfolgt durch die *Wissenserwerbskomponente*. Mit deren Hilfe lässt sich das vorhandene Expertenwissen in das System einpflegen. Die *Erklärungskomponente* unterstützt das Nachvollziehen des Ergebnisses durch Erläuterungen zu den getätigten Entscheidungen.

2.1.1 CAS Configurator Merlin Enterprise

Das Produkt CAS Configurator Merlin Enterprise ist die Konfigurationslösung der CAS Software AG für große Unternehmen. Das Produkt besteht aus einigen Standardkomponenten, die auf die einzelnen Bedürfnisse der Großkunden angepasst werden. In Abbildung 2.2 ist der Aufbau und das Zusammenspiel der verschiedenen Komponenten des Configurators zu sehen.

Die Wissensverarbeitungs-Komponente aus Abschnitt 2.1 ist hier der sogenannte *Konfigurationskern*. Er ist für die Auswertung der getätigten Auswahl von Elementen des Produktes zuständig. Mithilfe von booleschen Regeln, welche mit dem *Produktpflegeeditor* modelliert wurden und in einer *Datenbank* gespeichert sind, wird die Auswahl verifiziert.

Der Konfigurationskern berechnet ebenfalls sogenannte Alternativen. Diese treten auf, sobald die derzeitige Selektion alleine, ohne Hinzunahme von weiteren Bauteilen, nicht umsetzbar ist. Der Configurator kann in diesem Fall neue Möglichkeiten (Alternativen) vorschlagen, damit die Konfiguration durchgeführt werden kann. Die Auswahl der Konfigurationselemente erfolgt im sogenannten Configurator-Client. Der Client ist mit der Benutzerschnittstelle im Expertensystem zu vergleichen.

Der Konfigurationskern, sowie der Client befinden sich auf einem Java-Applikations-Server. Der Produktpflegeeditor ist eine eigenständige Rich-Client Anwendung, welche auf dem Eclipse Rich-Client-Plattform Framework[RCP] basiert.

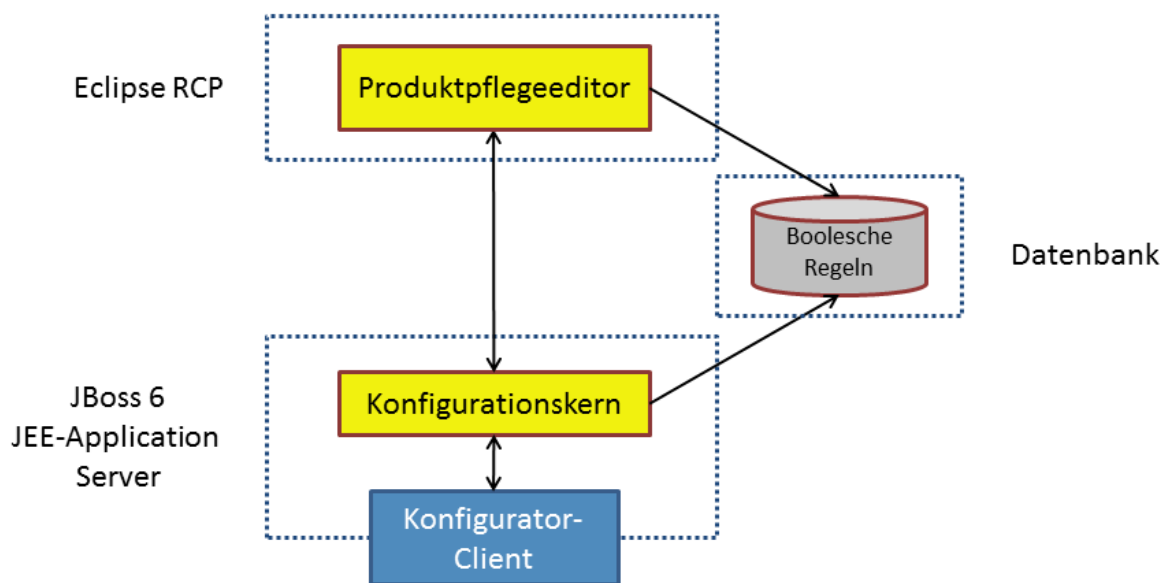


Abbildung 2.2: Architektur der Airbus Komponenten

2.1.2 Konfigurator des Kunden Airbus

Die Arbeit wird speziell für den Flugzeughersteller Airbus¹ angefertigt und beruht damit auf deren Konfigurationsmöglichkeiten. Airbus verwendet den Konfigurator für das Upgraden von bereits vorhandenen Flugzeugen einzelner Fluggesellschaften. Ein Beispiel für ein solches Upgrade können diverse Systemupgrades, wie bspw. ein neues Navigationssystem oder Ähnliches sein. Eine weitere Besonderheit im Konfigurationsprozess ist das Prüfen mit einzelnen Flugzeugen. Jedes verkaufte Flugzeug ist in der Datenbasis erfasst, sowie deren genaue Spezifikation und Bauteile. Der Konfigurator kann die Baubarkeit der einzelnen Upgrades genau auf ein bestimmtes Flugzeug prüfen lassen.

Bei der Modellierung von Konfigurationen werden sogenannte Status verwendet. Diese Status (engl. states) werden für Flugzeuge angelegt. Jeder einzelne Status repräsentiert eine erreichbare Konfiguration. Um einen Übergang zu einem state zu erreichen, wird ein bestimmtes Upgrade eingebaut

Für die Konfiguration wird z.Zt. eine Weboberfläche als Konfigurations-Client verwendet. Diese Oberfläche ist sehr einfach gehalten und kann nur von Experten bedient werden, da mit einzelnen Produktcodes bei der Konfiguration gearbeitet wird.

¹ <http://www.airbus.com/>

Die Weboberfläche befindet sich auf dem gleichen Applikationsserver wie der Kern und es besteht eine direkte Anbindung. In der Datenbank werden nicht nur Produkt- und Regelinformationen gehalten, sondern auch die Flugzeugdaten.

2.2 Grundlegende Anforderungen

Die Anwendung ist kein vom Kunden vorgegebenes Projekt. Dies hat zur Folge, dass es keine strikten Anforderungen an die App gibt. Die Anwendung soll, wie in Abschnitt 1.2 festgelegt den Konfigurationsprozess des Kunden vereinfachen und "mobil" verfügbar machen. Der Prototyp soll den Kunden überzeugen, seinen derzeitigen Konfigurationsprozess in den Neuen zu überführen. Die Anforderungen an die Software müssen demnach so festgelegt werden, dass sie für eine hohe Akzeptanz beim Kunden sorgt. Damit die Anwendung akzeptiert wird, müssen alle Gütekriterien dieses Ziel verfolgen.

Die Festlegung der Kriterien ist nach Rücksprache mit den Entwicklern und dem Projektleiter des Kunden entstanden. Gemeinsam wurde eine Priorisierung der einzelnen Anforderungen entsprechend der Zielsetzung erarbeitet. Im Folgenden ist das Ergebnis in Funktionale und Nicht-Funktionale Anforderungen eingeordnet. Fachliche Funktionen der Anwendung sind in den funktionalen Anforderungen enthalten. Diese wurden anhand der bestehenden Software und den Möglichkeiten des Konfigurator erstellt. Weiterhin wurden zusätzliche Anforderungen, die aufgrund der mobilen Umgebung notwendig sind, hinzugefügt. Bei den Nicht-Funktionalen Kriterien steht die Usability, das Anpassen an die Umgebung , sowie die Kundenbegeisterung im Vordergrund.

2.2.1 Abgrenzung der Anforderungen

Aufgrund der begrenzten Zeit im Rahmen dieser Arbeit wird die Anwendung prototypisch implementiert. Dies bedeutet, dass die Daten nicht vollständig sein müssen. Allerdings müssen die Daten in sich konsistent und auch an die Datenstruktur des Kunden angelehnt sein. Der Prototyp soll die Möglichkeiten einer mobilen, auf den Kunden zugeschnittenen, Konfigurationslösung zeigen. Für das Erreichen dieses Ziel werden zwei abgeschlossene Use-Cases bei innerhalb der Arbeit implementiert,

die alle Funktionen der Anwendung beinhalten. Die Auswahl dieser Use-Cases ist Teil dieser Arbeit und wird im nächsten Kapitel beschrieben.

Die bestehenden Funktionen des Konfigurationsservers werden verwendet. Es sind evtl. Anpassungen an die Reihenfolge der Aufrufe aufgrund der mobilen Zielumgebung notwendig. Um die Konfigurationsergebnisse empfangen zu können, soll eine einfache Webservicekommunikation mit dem Server stattfinden, die Kommunikationssicherung, wie Authentifizierung und Verschlüsselung ist nicht Teil der Arbeit. Dies soll bei einer späteren, vollständigen Implementierung berücksichtigt werden.

2.2.2 Nicht-Funktionale Anforderungen

NR.	Anforderung	Beschreibung	Details	Priorität
N1	Einfache Bedienung	Die Anwendung soll von unerfahrenen Benutzern schnell bedient werden können.	<p>Schwerpunkte der Softwareergonomie[ERGO]:</p> <ul style="list-style-type: none"> • Selbstbeschreibungsfähigkeit • Lernförderlichkeit • Erwartungskonformität 	A
N2	Schnelle Bedienung	Die Auswahl der einzelnen Komponenten soll möglichst schnell getätigt werden können.	Es muss für eine flüssige Navigation zwischen den einzelnen Seiten gesorgt werden. Der Benutzer sollte keine langen Wartezeiten beim Bedienen der Anwendung haben.	A
N3	Offline-Online Modus	Die Anwendung soll einen Offline- und Online-Modus enthalten	Es sollen Konfigurationsdaten offline zusammengestellt werden können, die später oder zeitgleich online (mit Konfigurationsserver) geprüft werden.	A
N4	Alleinstellungsmerkmal	Um den Kunden zu begeistern ist ein besonderes Feature, welches die Anwendung als Alleinstellungsmerkmal besitzt wünschenswert.	Es sollen besonders "aufregende" Features, die eine Technologie bietet in die Anwendung integriert und in einem sinnvollen Kontext genutzt werden.	B

2.2.3 Funktionale Anforderungen

NR.	Anforderung	Beschreibung	Details	Priorität
F1	Upgradeauswahl	Es sollen Upgrades für Flugzeuge auswählbar sein	Bevor die Auswahl stattfindet, soll es verschiedene Filtermöglichkeiten geben, wie beispielsweise der Flugzeugtyp oder die Art des Upgrades.	A
F2	Flugzeugauswahl	Es müssen Flugzeuge eines bestimmten Kunden auswählbar sein.	Es soll auch hier, analog zu B1 die Möglichkeit der Auswahl von Flugzeugen auf einen vorher gefilterten Datensatz geben.	A
F3	Konfiguration	Nach erfolgter Auswahl werden die Ergebnisse der Konfiguration dargestellt.	Kommunikation mit dem Konfigurationsserver, sowie die Darstellung der Ergebnisse auf dem Client	A
F4	Alternativen	Bei mehreren Möglichkeiten einer Auswahl sollen Alternativen ausgewählt werden können	Übersichtliche Darstellung von Alternativenauswahl.	A
F5	Speichern	Die getätigte Auswahl soll gespeichert werden	Nach Abschluss der Konfiguration soll diese lokal gespeichert werden, um ein erneutes Laden zu ermöglichen.	B
F6	Laden	Getätigte Konfigurationen müssen geladen werden können	Beim Start der Anwendung sollen getätigte Konfigurationen geladen und angezeigt werden können	B

3 Konzeption der Anwendung

Im vorigen Kapitel wurden die Anforderungen, sowie das Umfeld der Arbeit erläutert. In diesem Abschnitt wird der aktuelle Workflow beim Konfigurieren analysiert und anschließend an die mobile Umgebung angepasst. Nach der Workflow-Modellierung wird mit einer Entscheidung über den richtigen Typ der App.

3.1 Aktueller Konfigurationsprozess

Im aktuellen Prozess des Kunden wird ein Upgrade zuerst über den Produktkatalog gefunden. Im Katalog ist eine eindeutige Nummer enthalten, die bei der Bestellung verwendet wird. Die Fluggesellschaft nennt zu der Upgrade-Nummer, die Flugzeuge, die dieses Upgrade erhalten sollen. Diese Informationen werden von einem Design-Manager angenommen. Für die Weiterverarbeitung werden diese Informationen im Konfigurator-Client aus Abschnitt 2.1.2 erfasst. Die einzelnen Schritte der Erfassung sind in Abbildung 3.1 zu sehen.

Im ersten Schritt wird das passende Flugzeugprogramm ausgewählt. Ein Programm ist eine grobe Einteilung für Flugzeuge nach deren Größe und Art. Die Auswahl ist eine erste Filterung der Datensätze. Des weiteren wird mit Hilfe des Programms das Regelwerk, welches auf dem Konfigurationsserver verwendet wird festgelegt. Anschließend folgt die Auswahl der entsprechenden Flugzeuge, welche ein Upgrade erhalten sollen. Die Auswahl kann nach bestimmten Kriterien gefiltert werden. Bei der Bestellung eines Upgrades wird die Flugzeugnummer angegeben, womit eine schnelle Auswahl erfolgen kann. Sind die Flugzeuge ausgewählt, werden Upgrades aus einer Liste selektiert. Es sind alle verfügbaren Updates aufgelistet. Die Auswahl erfolgt ebenfalls mit der eindeutigen Nummer, welche bei der Bestellung angegeben wurde.

Nach der letzten Auswahl sind alle für die Konfiguration benötigten Elemente ausgewählt. Es folgt eine Validierung der Flugzeuge. Bei dieser Überprüfung werden die

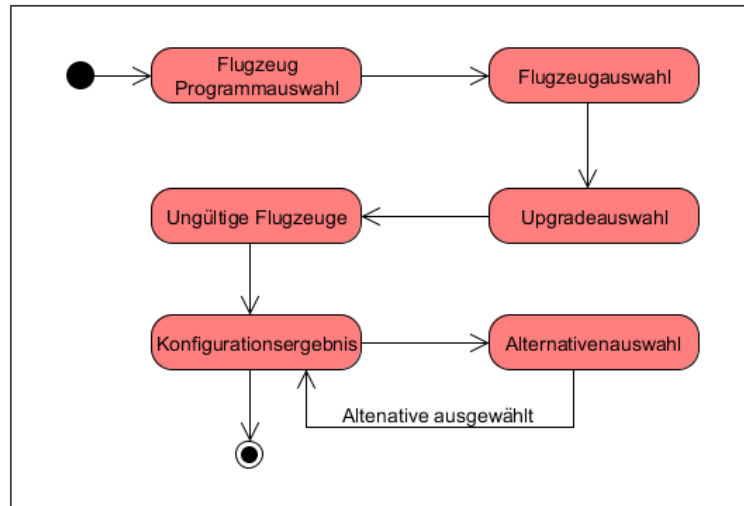


Abbildung 3.1: Programmablauf des Konfigurations-Clients

einzelnen Flugzeuge auf Konfigurationen untersucht, die in Widerspruch mit dem ausgewählten Upgrade stehen. Wenn keine Widersprüche gefunden wurden, ist die Bildung von sogenannten Konfigurationsgruppen die nächste Aufgabe. Eine Konfigurationsgruppe enthält Flugzeuge, die in die gleichen Zielzustände kommen, wenn das Upgrade eingebaut wird. Wenn es mehrere Möglichkeiten gibt, um in einen bestimmten Zustand des Flugzeuges zu kommen, werden sogenannte Alternativen in einer Konfigurationsgruppe enthalten. Damit die Konfiguration vollständig ist, muss der Anwender für die Gruppe eine Alternative auswählen.

Nachdem eine vollständige Konfiguration erzeugt wurde, wird daraus ein Excel-Dokument generiert. In diesem sind die Upgrades enthalten, die in den einzelnen Flugzeuge eingebaut werden müssen. Aus dem Dokument wird ein Upgrade-Angebot erstellt, dass anschließend dem Kunden vorgelegt wird.

3.2 Workflow Modellierung

Beim derzeitigen Konfigurationsprozess wird die eigentliche Konfiguration dem Experten überlassen. Ein Kunde wählt die Codes und Identifikationsnummern aus Katalog undzeitigem Flugzeugbestand aus, erhält jedoch erst nach der Arbeit des Experten eine Bestätigung über die Gültigkeit der Konfiguration. Dieser Prozess wird bei der Portierung auf die mobile Umgebung verändert, so dass ein neuer Workflow

entsteht. Die Prozesse sind im Folgenden in einen mobilen Konfigurationsprozess und den unterstützenden App-Workflow unterteilt.

3.2.1 Mobiler Konfigurationsprozess

Der derzeitige Workflow sollte die neuen Möglichkeiten einer mobilen Konfigurationsumgebung beinhalten. Mit der Mobilität ist ein Szenario einer direkten Konfiguration mit dem Kunden und einem Vertriebsexperten möglich. Beide können mithilfe der App direkt kommunizieren und das Upgrade gemeinsam durchführen. Primäres Ziel dieser Anpassung ist den Prozess kundenfreundlicher zu gestalten. Dieses Ziel soll durch folgende zwei Maßnahmen erreicht werden:

- **Vereinfachung der Auswahl:** Das An- und Abwählen der Flugzeuge, bzw. der Upgrades soll vereinfacht werden. Die Auswahl soll nicht nur durch die Produktcodes, sondern durch verständlichere Weise erfolgen.
- **Schnelleres Feedback:** Durch eine mobile Lösung soll bereits beim Kunden ein Feedback über die Gültigkeit der Konfiguration vorhanden sein.

Mit beiden Maßnahmen wird der Kunde stärker in den Prozess der Konfiguration einbezogen. Im Idealfall kann der Kunde die Anwendung alleine bedienen und der Experte steht nur Beratend zur Seite. Eine weitere Besonderheit der neuen Anwendung kann sein, dass der Kunde zu einem weiteren Upgrade bewogen wird, welches er erst beim Benutzen der Anwendung entdeckt. Eine weitere Möglichkeit ist das gezielte Anzeigen von kundenspezifische Informationen, die für weitere Kaufaktivitäten sorgen können.

3.2.2 Workflow der App

Der Anwendungsworkflow muss zu dem neuen fachlichen Prozess passen und ihn unterstützen. Dies geschieht durch Anpassungen an die Kommunikation mit dem Konfigurationsserver. Der aktuelle Client und der Server befinden sich auf der gleichen Umgebung. Somit können beide direkt miteinander kommunizieren und benötigen keine aufwändigen Webservice Schnittstellen. Für den neuen Prozess muss der Konfigurationsserver von der mobilen Umgebung erreichbar sein. Dies bedeutet, dass es eine Schnittstelle für die App geben muss. Die Kommunikation mit dem Webserver

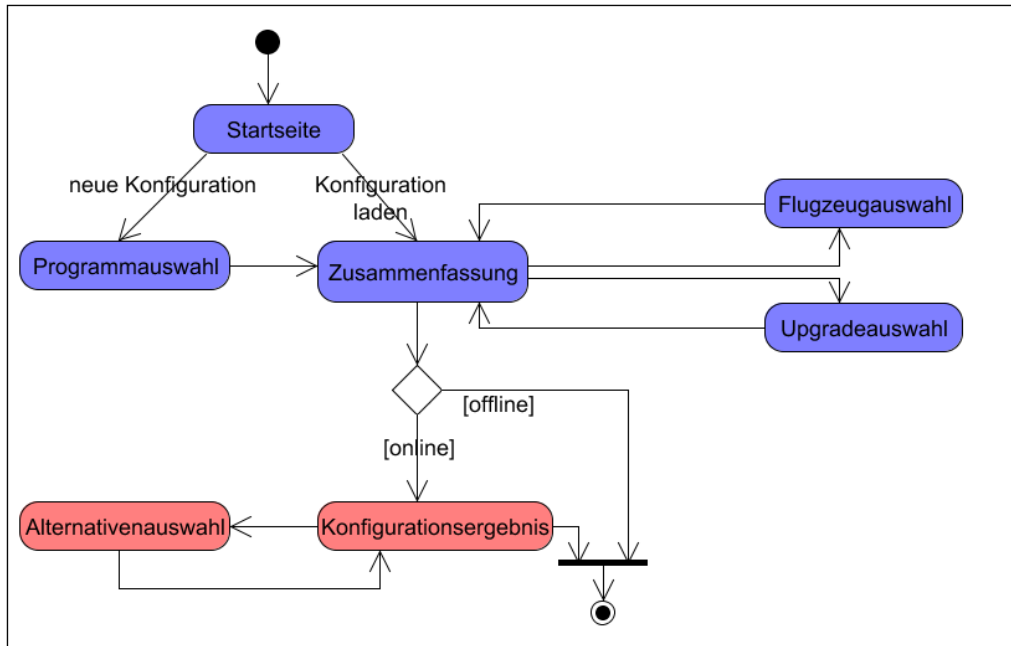


Abbildung 3.2: Programmablauf der Konfigurator-App

kann durch den mobilen Einsatz erschwert werden. Dies ist beispielsweise der Fall, wenn die Konfiguration beim Kunden vor Ort stattfindet und eine schlechte, bis gar keine Verbindung vorhanden ist. Aus diesem Grund ist bei den Nicht-Funktionalen Anforderung in Abschnitt 2.2.2 ein sogenannter Offline Modus enthalten, der die Zusammenstellung einer Konfiguration auch ohne Anbindung an den Konfigurator ermöglicht. Um eine flüssige Navigation gewährleisten zu können, muss die Kommunikation mit dem Server auf das Wesentliche konzentriert sein. Dies bedeutet, dass eine Konfiguration erst am Ende stattfindet und die einzelnen Schritte, die im derzeitigen Client durchgeführt werden zusammengelegt werden müssen.

In Abbildung 3.2.2 ist der Anwendungsverlauf der App zu sehen. Analog zu der Weboberfläche wird bei einer neuen Konfiguration zuerst ein Programm (**Programmauswahl**) ausgewählt. Nach der Auswahl gelangt der Benutzer in eine **Zusammenfassung** der aktuellen Konfiguration. Von dieser Ansicht aus können Flugzeuge (**Flugzeugauswahl**) oder Upgrades (**Upgradeauswahl**) selektiert werden. Sobald beides vorhanden ist, wird die Erreichbarkeit der Konfigurationsschnittstelle geprüft. Bei Verfügbarkeit befinden wir uns im "Online" Modus und es werden die Konfigurationsgruppen angezeigt. Nach erfolgter Konfiguration werden die Alternativen in einer separaten Ansicht (**Alternativenauswahl**) ausgewählt. Ist die Konfiguration vollständig, so kann

das Upgrade direkt bestellt werden und den Update Vorgang beenden. Ist der Server nicht verfügbar, kann der Benutzer die aktuelle Konfiguration speichern und damit den Prozess abschließen. Anschließend kann die Auswahl direkt über den Startbildschirm geladen werden.

Um für einen möglichst geringen Kommunikationsaufwand mit dem Konfigurationsserver zu sorgen, wird der Server erst nach erfolgter Auswahl bedient. Der Typ bezieht sich auf eine Art Die Kommunikation wird alle zuvor beinhalteten Einzelschritte auf dem aktuellen Client zu einem Zusammenfügen. Die Trennung von Online (rot) und Offline (blau) ist in der Abbildung ebenfalls aufgezeigt.

3.3 Auswahl der Anwendungsplattform

Nachdem der Workflow, sowie die Anforderungen der Anwendung erläutert wurden, folgt in diesem Abschnitt die Auswahl der passenden Anwendungsplattform. Es werden die drei verschiedenen Anwendungstypen nativ, web und hybrid untersucht [PLATTFORM]. Bei jeder dieser Plattformen werden die bekanntesten Technologien verwendet. Die Untersuchung beruht auf den in Kapitel 2.2 gestellten Anforderungen. Es werden folgende Auswahlkriterien verwendet:

- **Offlinemodus:** Können offline Daten beliebig gespeichert werden? Gibt es eine passende Technologie für einen solchen Modus?
- **Sicherheit:** Da bei der Konfiguration sensible Daten verwendet werden, muss die Anwendung entsprechend gesichert werden können. Die einzelnen Sicherungsmaßnahmen sollen hierbei untersucht werden.
- **Schnittstelle:** Welche Möglichkeit für eine Schnittstelle zu dem Konfiguratorserver gibt es?
- **Kundenpräferenz:** Gibt es eine bestimmte Technologie/Plattform, die vom Kunden explizit gewünscht ist? Hat die Technologie ein besonderes Alleinstellungsmerkmal?
- **Performance:** Für die Anzeige der einzelnen Produktbilder, sowie eine schnelle Navigation innerhalb der Anwendung ist die Performance des jeweiligen Typs wichtig.

Bei der Auswahl der Technologien werden nur Plattformen ausgewählt, die auf Tabletgeräten verfügbar sind. Es werden alle Kriterien für den Anwendungstyp bewertet und zusätzlich für die einzelnen Technologien.

3.3.1 Native Anwendungen

Native Anwendungen sind auf die jeweilige Zielplattform beschränkt. Dieser Typ einer Anwendung beschränkt sich auf das zuvor gewählte Betriebssystem. Der geschriebene Programmcode kann meistens nur auf der Zielplattform verwendet werden. Für die Bewertung von nativen Anwendungen werden die drei am Meisten genutzten Betriebssysteme im Tabletbereich verwendet: Android, iOS und Windows 8[NATIVE].

Offlinemodus: Bei einer nativen App ist keine Serveranbindung notwendig. Alle Daten können ohne eine Internetverbindung lokal auf dem Gerät selbst vorhanden sein. In einer nativen Anwendung können alle Dateioperationen, wie auf einem normalen Betriebssystem durchgeführt werden. Dies hat den großen Vorteil, dass auf den Speicher zugegriffen werden kann. Große Datensätze, die beim Offlinemodus anfallen, können damit auf dem Gerät gespeichert werden. In diesem Kriterium ist das Verhalten für alle drei Betriebssysteme gleich, womit keine Unterscheidung der einzelnen Technologien notwendig ist.

Sicherheit: Durch die native Umgebung ist die Anwendung nur auf dem Betriebssystem vorhanden und in einer geschlossenen Umgebung. Die Kommunikation mit dem Webserver muss allerdings selbstständig verschlüsselt werden, insbesondere wenn die Anwendung bei einem Kunden in dessen Netz verwendet wird. Ein weiteres Problem, welches native Anwendungen besitzen sind die lokalen Sicherheitslücken eines jedes Betriebssystems. Regelmäßig tauchen neue versteckte Fehler im Betriebssystem auf, die Schaden zufügen könnten.

- **Android und iOS:** Bei beiden Betriebssystemen werden regelmäßig Sicherheitslücken offengelegt. Die Angriffe auf beide Systeme ist zunehmend, wobei das iOS-System im Vergleich besser abschneidet und in mehreren
- **Windows 8:** Durch die hohe Verbreitung von Windows Betriebssystemen sind diese Systeme besonders anfällig, bzw. oft verwendete Ziele bei Angriffen. Die breite Anzahl der Nutzer sorgt jedoch für ausreichend Lösungen zur Sicherung der Anwendung.

Schnittstelle: Für die Kommunikation mit dem Konfigurationsserver bieten sich bei einer nativen Anwendung Webservices an. Diese können in den Standards

3.3.2 Web Anwendungen

phone gapp, gwt,

3.3.3 Hybride Anwendungen

3.3.4 Abwägung

4 Entwurf der Benutzeroberfläche

4.1 Windows 8

4.1.1 Bedienkonzepte

4.1.2 MVVM

4.2 Design der Ansichten

4.3 Interaktion der Ansichten

5 Implementierung

5.1 Implementierung des Katalog-Workflows

5.2 Implementierung des Konfigurations-Workflows

6 Evaluation der Anwendung

6.1 Zielkriterien

6.2 Testergebnisse

7 Fazit und Ausblick

Abbildungsverzeichnis

2.1	Aufbau eines Expertensystems [KELLER, s.6]	4
2.2	Architektur der Airbus Komponenten	6
3.1	Programmablauf des Konfigurations-Clients	12
3.2	Programmablauf der Konfigurator-App	14

Listings

Literaturverzeichnis

[MASS] PROF. DR. DANIEL MARKGRAF **Wirtschaftslexikon** Mass Customization, <http://wirtschaftslexikon.gabler.de/Definition/mass-customization.html>, aufgerufen am 27.05.2013

[PUPPE] FRANK PUPPE **Springerverlag** Einführung in Expertensysteme, 1. Aufl., 1988

[KELLER] HUBERT B. KELLER **DH Karlsruhe** Wissensbasierte Systeme - Einführung, Vorlesung SS 2007

[EXPERT] GERALD REIF **Deutsches Forschungszentrum für Künstliche Intelligenz** Expertensysteme, <http://www.dfki.uni-kl.de/aabecker/Mosbach/Experten/Reif-node8.html>, aufgerufen am 18.07.2013

[RCP] RALF EBERT Eclipse RCP, http://www.ralfebert.de/eclipse_rcp/EclipseRCP.pdf, Version 1.1, 19.08.2011

[ERGO] CHRISTIANE RUDLOF **Unfallkasse Post und Telekom** Handbuch Software-Ergonomie. Useability Engineering., <http://www.ukpt.de/pages/dateien/software-ergonomie.pdf>, S.52, 2. Auflage, Tübingen 2006

[PLATTFORM] CLOUDSHERPAS Native, Hybrid and Mobile Web Apps, <http://www.cloudsherpas.com/services/custom-development/mobile-apps/native-hybrid-and-mobile-web-application-development/>, aufgerufen am 23.07.2013

[NATIVE] STATISTA Apple bleibt Marktführer, https://www.bsi.bund.de/cae/servlet/contentblob/476464/publicationFile/30642/WebSec_pdf, aufgerufen am 23.07.2013

- [WEBAPP1] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATI-
ONSTECHNIK **SecureNet** Sicherheit von Webanwendungen,
<http://de.statista.com/themen/580/tablets/infografik/758/prognose-der-weltweiten-marktanteile-der-tablet-betriebssysteme/>, 1. Version, August 2006
- [WIN8-1] BART CLAEYS, QIXING ZHENG **MSDN** Designfallstudie:
vom iPad zur Windows Store-App, <http://msdn.microsoft.com/de-de/library/windows/apps/hh868262>, aufgerufen am 16.05.2013
- [WIN8-2] MICROSOFT **MSDN** Entwerfen großartiger Produktivitäts-Apps für Win-
dows, <http://msdn.microsoft.com/de-de/library/windows/apps/hh868273>,
aufgerufen am 16.05.2013
- [WIN8-3] MICROSOFT **MSDN** Shopping-Apps, <http://msdn.microsoft.com/de-de/library/windows/apps/jj635241.aspx>, aufgerufen am 16.05.2013
- [WIN8-4] MICROSOFT **MSDN** am 16.05.2013 Planen ihrer App,
<http://msdn.microsoft.com/de-de/library/windows/apps/hh465427>, auf-
gerufe