

Eine Multikanal-Architektur für adaptive, webbasierte Frontendsysteme und deren Erweiterbarkeit durch Variantenbildung

Dipl.-Inform. Michael Thomas Hitz
DHBW-Stuttgart + Allianz Deutschland AG
Paulinenstraße 50
70178 Stuttgart
hitz@dhbw-stuttgart.de

Abstract: Der in den letzten Jahren stark gestiegene Bedarf an webbasierten Zugängen zu den Systemen eines Unternehmens für unterschiedliche Nutzergruppen (fachliche Kanäle) führte häufig dazu, dass parallel *monolithische, siloartige* Frontend-Systeme entstanden sind, welche in Wartung und Weiterentwicklung durch die resultierenden Redundanzen hohe Kosten verursachen. In den letzten Jahren rückten zudem neue Technologien wie mobile Geräte weiter in den Fokus, die neben den fachlichen Kanälen eigene Anforderungen an die Darstellung und die Prozesse haben und damit die Komplexität weiter erhöhen.

Um schnell und kosteneffizient auf den Markt reagieren zu können, bedarf es einer Lösung, welche Funktionalitäten kanalübergreifend wiederverwendbar macht und bei Änderungen oder kanalspezifischen Erweiterungen Redundanzen durch Bildung von Varianten vorhandener Oberflächen und Prozesse vermeidet.

1 Einleitung

1.1 Motivation

Die letzten Jahrzehnte der Softwareentwicklung im Web-Umfeld produzierten Frontend-Systeme¹, die spezifisch auf Nutzergruppen (**fachliche Kanäle**) zugeschnitten wurden. Diese Fokussierung der in den entstandenen Portalen entwickelten Anwendungen auf einen bestimmten Kanal führte in vielen Fällen dazu, dass Oberflächen und Anwendungsabläufe mehrfach entwickelt wurden, obwohl diese häufig für die fachlichen Kanäle gleich sind oder in nur leicht modifizierter Form wiederverwendbar wären.

Mit der stetig wachsenden Verbreitung neuer Technologien wie z.B. Spracherkennung und intelligenterer mobiler Geräte müssen weitere Anforderungen durch die Oberflächen und

¹Der Begriff (*webbasiertes Frontend-System*) wird hier als Summe aller Komponenten verwendet, die notwendig sind, eine Anwendung dem Nutzer zu präsentieren. Dies beinhaltet Oberflächen, Seitenfluss und Anwendungslogik, welche den Zugriff auf Prozesse in den Backendsystemen orchestriert. Die Backendsysteme werden dabei als bestandsführend betrachtet und stellen die eigentlichen Businessfunktionalitäten bereit - und sind damit nicht Teil des Frontend-Systems.

Anwendungsabläufe erfüllt werden (z.B. darstellbare Datenmenge, eingeschränkte Konnektivität). Aus Systemsicht sind dies weitere Kanäle (**technische Kanäle**), für die bisher meist ebenfalls eigene Portale parallel aufgebaut wurden.

Die Entstehung solcher monolithischen *Entwicklungs-Silos* hat sowohl organisatorische als auch technische Ursachen ([Ban01]). Organisatorische Gründe sind dabei insbesondere die Organisationsstruktur der IT in größeren Unternehmen, die meist entlang der fachlichen Kanäle erfolgt und damit eine isolierte Softwareentwicklung begünstigt. Technische Ursachen liegen beispielsweise in der mangelnden Unterstützung des Szenarios durch die vorhandene Infrastruktur und Frameworks.

Die Redundanzen, die in der Systemlandschaft entstanden, sind nur schwer zu verwalten und verursachen in der Wartung und Erweiterung hohe Kosten. Eine Lösung ist die Schaffung eines multikanalfähigen Systems², in welchem Varianten zur Unterstützung neuer Kanäle schnell und zu geringen Kosten integriert werden können.

1.2 Ziel und Durchführung der Arbeit

Ziel der Arbeit wird es sein, einen Blueprint zu einer adaptiven, multikanalfähigen Frontend-Architektur und Vorgehensweisen zu erarbeiten, welche die folgenden Eigenschaften aufweisen:

- ein Basismodell, welches die Nutzung vorhandener Elemente über mehrere Kanäle hinweg vorsieht
- eine einfache und schnelle Integrierbarkeit neuer Kanäle (fachlich und technisch) und deren Spezifika durch Adaption und Variantenbildung von Oberflächen und Prozessen im Frontend gestattet

Hierzu werden anhand eines Szenarios aus dem Versicherungsumfeld Anforderungen und Eigenschaften abgeleitet, die ein Multikanalsystem aufweisen muss. Darauf aufbauend wird ein Basismodell für eine Architektur entwickelt, welches eine grundsätzliche Schichtung des Systems vorschlägt und über die Verantwortlichkeiten der Schichten eine detailliertere Betrachtung der Kernelemente gestattet.

Basierend hierauf fokussiert die Arbeit im Weiteren auf die Erforschung und das Design von Mechanismen, wie Variantenbildung auf Prozessebene erreicht und darauf aufbauend kanalspezifische graphische Oberflächen³ (automatisch oder ebenfalls durch Variantenbildung bestehender Abläufe) abgeleitet werden können. Das Ziel ist hierbei, eine hohe Wiederverwendung durch Variantenbildung bestehender Anwendungen zu erreichen.

²Im Rahmen dieser Arbeit wird unter dem Begriff *Multikanalsystem* ein System verstanden, das in der Lage ist, unterschiedliche fachliche und technische Kanäle einheitlich zu bedienen, indem durch dessen Aufbau ein hoher Grad an Wiederverwendung erreicht wird und trotzdem die Spezifika der einzelnen Kanäle abgebildet werden können.

³Den Schwerpunkt der Untersuchungen bilden hierbei webbasierte Oberflächen. Ziel ist jedoch auch die Anwendbarkeit der Lösungen auf Desktop-Anwendungen, die hinsichtlich der Präsentation neben webbasierten Browser- und mobilen Anwendungen weitere Anforderungen stellen.

1.3 Forschungsfragen

Die daraus resultierenden Fragen für die weiteren Forschungen sind

- Welche Anforderungen leiten sich aus der Forderung nach *Multikanalfähigkeit* eines Systems ab und wie lassen sich diese Anforderungen in einem Basismodell einer Architektur erfüllen?
- Wie können Prozesse multikanalfähig definiert und durch Variantenbildung die Spezifika weiterer Kanäle berücksichtigt werden?
- Wie können kanalspezifische Oberflächen in einem Multikanalsystem für diese Prozesse erstellt werden, sodass schnell und kostengünstig Erweiterungen erfolgen können?

1.4 Methodik

Zur Analyse der Anforderungen an ein Multikanalsystem wird ein Szenario aus dem Versicherungsbereich gewählt, welches fachlich durch strukturierte Interviews mit einem oder mehreren am Forschungsprojekt beteiligter Partner erstellt wird. Zudem erfolgt eine IST-Analyse auf Architekturebene, um daraus ggf. weitere, technische Anforderungen und Problemstellungen ableiten zu können.

Darauf aufbauend und unter Einbeziehung relevanter Forschungsergebnisse und Literatur in diesem Umfeld, wird ein Basismodell für eine Multikanalarchitektur hergeleitet. Dieses soll mittels geeigneter Vorgehensweisen hinsichtlich seiner Qualität bewertet und auf die Tragfähigkeit geprüft werden. Hierzu existieren in der Literatur unterschiedliche Ansätze, von denen ein geeignetes Verfahren ausgewählt wird (z.B. [KLKB07]).

Die Forschungen im Bereich der Oberflächen und Prozesse und technischen Herausforderungen gründen auf Forschungsergebnissen, welche Teilaspekte der Aufgabenstellung behandeln. Die Verifikation der im Rahmen des Projekts erarbeiteten Ergebnisse erfolgt durch prototypische Umsetzung bzw. realem Einsatz bei am Projekt beteiligter Partner (in Zusammenarbeit mit der DHBW Stuttgart, der Hochschule für Technik Stuttgart und der Allianz Deutschland AG).

2 Verwandte Arbeiten

Es existiert eine Reihe von Arbeiten, die architekturelle Aspekte (multikanalfähiger) Anwendungen untersuchen (z.B. [BGK⁺97], [Per06], [ZDGH05], [YN08]), zu Themenstellungen hybrider Oberflächentechnologien und zu Konzepten zur Lösung technischer Herausforderungen in diesem Umfeld (z.B. [BGL06],[JrTD04], [MP02]). Auf Ebene serviceorientierter Architekturen und Prozessmanagements existieren Forschungen im Bereich mandantenfähiger und adaptiver Prozesse (z.B. [WKNL07], [DR09]), die einen grundlegenden Beitrag leisten.

Die vorgeschlagene Arbeit soll die Forschungen im Bereich multikanalfähiger Systeme um den Themenbereich der Erweiterbarkeit durch Variantenbildung auf Oberflächen- und Prozessebene ergänzen und sie im Rahmen eines Multikanalsystems zusammenführen.

3 Erste Ergebnisse - Stand der Arbeit

Die Arbeit befindet sich in der Phase der Vorstudien, welche die Machbarkeit bereits vorhandener Lösungsideen für die oben angesprochenen Forschungsfragen untersuchen.

Grundlage hierfür waren erste Analysen im Versicherungskontext für Multikanalsysteme relevanter Nutzergruppen. Als erstes Szenario aus dem Versicherungsbereich wurde das Thema „Tarifizierung mit Antrag“ gewählt, da es sich hier um geschlossene Prozesse handelt, die von mehreren Kanälen mit leichten Variationen durchgeführt werden.

3.1 Erstes Basismodell

Das hergeleitete Basismodell entstand unter der Annahme, dass mehrere fachliche Kanäle mit unterschiedlichen Geräten das webbasierte System nutzen. Das System soll für alle Kanäle einheitlich sein, sodass dieselbe Infrastruktur verwendet werden kann. Basierend auf den bisherigen Erfahrungen in der Portalentwicklung und der grundsätzlichen Entscheidung, einen serviceorientierten Ansatz bei der Integration der Backend-Systeme zu verwenden, ergab sich das in Abbildung 1 dargestellte Basismodell. Es folgt auf der Businesslogik-Ebene den in [BGK⁺97] angeführten Grundsätzen und konkretisiert die technischen Applikationsaspekte.

Den *kanalspezifischen Einstiegspunkt* bildet die mit *Platform/Application Integration* bezeichnete Ebene, die für die Bereitstellung und Aufbereitung kanalspezifischer Informationsinhalte und Integration der Applikationen zuständig ist. Die Applikationsschicht (*Application Layer*) beinhaltet die Oberflächensteuerung der Anwendungen für unterschiedliche fachliche und technische Kanäle. Auf Ebene des (*Business Orchestration Layer*) leben die Prozesse, die durch das Frontend orchestriert werden. Diese sind nach Anwendungsfall (z.B. „Tarifizierung einer Lebensversicherung“) und nach Kanal (z.B. „Außendienst“) gegliedert. Es ist sinnvoll, auf dieser Ebene *Business Process Engines* zu verwenden, da diese zu einfacher anpassbaren Systemen führen. Hier werden Services des *Business Services Layer* genutzt, welche den Zugang zur eigentlichen Business-Logik und den Daten in den Backendsystemen regeln.

Diskussion: Essentiell bei diesem Modell ist, dass die Schichten voneinander entkoppelt sind und so eine Variabilität möglich ist. Eine übergeordnete Schicht kann durch eine alternative Lösung ersetzt oder um eine neue Variante ergänzt werden. Dadurch wird die Funktionalität der unteren Schicht wiederverwendet. Dies ist insbesondere für die Integration neuer Kanäle wichtig. Mit diesem Modell lassen sich die von den Schichten zu erfüllenden Aufgaben und verbundene Multikanalanforderungen, aber auch infrastrukturelle Fragen (Application Server, BPM System, CMS, etc.) diskutieren.

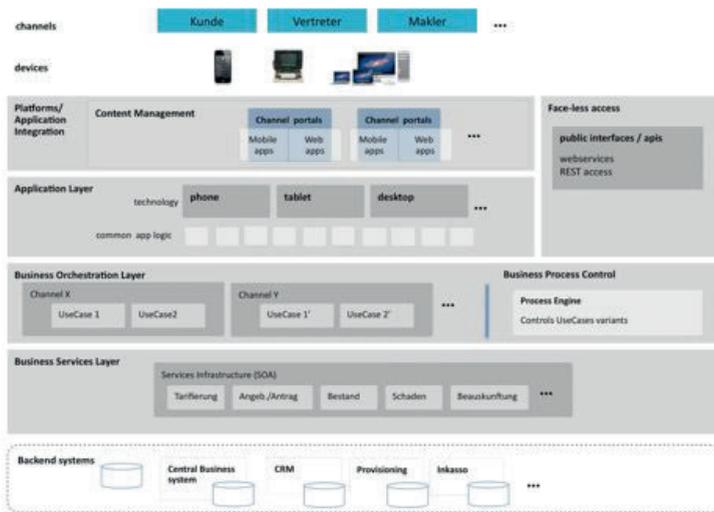


Abbildung 1: Basismodell für eine Multikanalarchitektur

3.2 Lösungsidee zur Variantenbildung

Das gefundene Basismodell legt eine nähere Untersuchung zur Variantenbildung insbesondere auf dem *Application Layer (AL)* und dem *Business Orchestration Layer (BOL)* nahe, da hier die (kanalspezifischen) Anpassungen häufiger vorkommen (s. auch [BGK⁺97]). Auf dem BOL kann eine Variantenbildung dadurch erreicht werden, dass Prozesse zu einem Anwendungsfall in *Basis-* und *kanalspezifischen* Anteil gegliedert werden. Der Basisprozess wird mit Erweiterungspunkten (EP) versehen, an denen kanalspezifische zusätzliche Prozessschritte eingehängt werden können. Für Oberflächen wird entgegen den modellierenden Ansätzen in Konzepten wie z.B. WebML ([CDM03]) eine automatisierte Erzeugung verfolgt (Ansätze hierzu in [BGL06]), welche mit den Eigenschaften der im Rahmen der Prozesse bearbeiteten Daten arbeitet. Wir gehen davon aus, dass auf der Obermenge der Daten aller Teilprozesse mit weiteren Eigenschaften versehen (Datenkohäsion, Kanalrelevanz, . . .) die Oberflächen kanalspezifisch erzeugt werden können.

3.3 Untersuchungen zu Oberflächen in multikanalfähigen Anwendungen

Das Ziel war es, einen ersten Schritt für die Oberflächen-Konzepte zu tun, der technologie-neutrale, kanalspezifische Dialoge bereitstellt. Im ersten Schritt wird durch eine technologie-neutrale, qualitative Beschreibung der Oberflächen die einfache Wiederverwendung in unterschiedlichen Kontexten gezeigt. Für die Machbarkeitsanalyse fand hier eine Be-

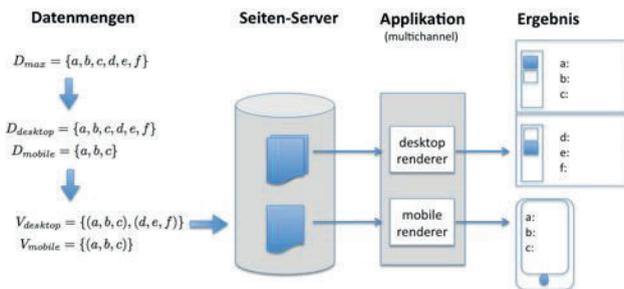


Abbildung 2: Oberflächen in einer Multikanalanwendung

schränkung auf die technischen Kanäle *mobile* und *desktop* statt.

Das Konzept basiert auf der Annahme, dass ein Prozess (hier Tarifierungsbaustein) über alle Kanäle hinweg auf einer Maximalmenge an Daten operiert (D_{max}). D_{max} ist bestimmt von der Summe der Eingabedaten, welche von den im Prozessverlauf verwendeten Backendservices benötigt werden. Unterschiedliche Kanäle arbeiten auf Untermengen dieser Menge ($D_{desktop}$, D_{mobile}), indem sie nur Teile dem Benutzer präsentieren (z.B. werden auf mobilen Geräten üblicherweise weniger Daten vom Benutzer eingefordert als bei Desktopbrowser-Oberflächen). Diese Untermengen werden zudem für Kanäle auf unterschiedlichen Dialogseiten/Sichten gruppenweise abgefragt bzw. dargestellt ($V_{desktop}$, V_{mobile}). Ergänzt um Typ- und allgemeine Darstellungsinformationen können diese Seitenbeschreibungen nun zum Aufbau der Dialoge in einer GUI-Technologie herangezogen und für das Endgerät gerendert werden (s. Abbildung 2).

Diskussion: Mit einem der gewählten UI-Technologie entsprechenden Renderer gestattet es der Ansatz, einfach Oberflächen für neue Kanäle zu definieren oder Bestehende für unterschiedliche technische Kanäle zu verwenden. Die Operationen, die ggf. mit Oberflächenelementen verknüpft sind (Buttons etc.) müssen damit umgehen können, dass nur Subsets der Daten zur Verfügung stehen oder kanalspezifische Implementierungen an die Oberflächen gebunden werden. Der Ansatz wurde bereits prototypisch umgesetzt und für client-seitige UI-Technologien (im Prototyp Sencha ExtJS/touch) verifiziert.

4 Ausblick

Die bisherigen Untersuchungen im GUI-Bereich sind vielversprechend. Hier muss eine weitere Verfeinerung hin zur Lösungsidee erfolgen und der bisherige Schritt der „Seitendefinition“ mittels Wissen über die Beziehungen der Daten untereinander etc. durch weitergehende Automatisierung ersetzt werden. Das Thema der Variantenbildung bei Prozessen ist zu untersuchen. Zudem müssen die Konzepte in die Gesamtarchitektur eingebettet werden.

Literatur

- [Ban01] Frank Bannister. Dismantling the silos: extracting new value from IT investments in public administration. *Information Systems Journal*, 11:65–84, 2001.
- [BGK⁺97] Dirk Bäumer, Guido Gryczan, Rolf Knoll, Carola Lilienthal, Dirk Riehle und Heinz Züllighoven. Framework Development. *Communications of the ACM*, 40(10):52–59, 1997.
- [BGL06] Matthias Book, Volker Gruhn und Matthias Lehmann. Automatic dialog mask generation for device-independent web applications. In *Proceedings of the 6th international conference on Web engineering - ICWE '06*, Seiten 209–216, New York, New York, USA, 2006. ACM Press.
- [CDM03] Stefano Ceri, Florian Daniel und Maristella Matera. Extending WebML for Modeling Multi-Channel Context-Aware Web Applications. In *Fourth International Conference on Web Information Systems Engineering Workshops, 2003. Proceedings.*, Seiten 225–233, 2003.
- [DR09] Peter Dadam und Manfred Reichert. The ADEPT project: a decade of research and development for robust and flexible process support. *Computer Science - Research and Development*, 23(2):81–97, April 2009.
- [JrTD04] Ivar Jø rstad, Do Van Thanh und Schahram Dustdar. Towards Service Continuity for Generic Mobile Services. In *IFIP International Conference on Intelligence in Communication Systems (INTELLCOMM 04)*, 2004.
- [KLKB07] Chang-ki Kim, Dan-hyung Lee, In-young Ko und Jongmoon Baik. A Lightweight Value-based Software Architecture Evaluation. In *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, Seiten 646–649, 2007.
- [MP02] Guido Menkhaus und Wolfgang Pree. A Hybrid Approach to Adaptive User Interface Generation. In *Proceedings of the 24th International Conference on Information Technology Interfaces, 2002. ITI 2002.*, Seiten 185–190, 2002.
- [Per06] Barbara (Ed.) Pernici. *Mobile Information Systems - Infrastructure and Design for Adaptivity and Flexibility*. Springer, 2006.
- [WKNL07] Matthias Wieland, Oliver Kopp, Daniela Nicklas und Frank Leymann. Towards Context-aware Workflows. In *CAISE'07 Proceedings of the Workshop and Doctoral Consrtium*, Seiten 1–15, 2007.
- [YN08] Takuto Yanagida und Hidetoshi Nonaka. Architecture for Migratory Adaptive User Interfaces. In *8th IEEE International Conference on Computer and Information Technology, 2008. CIT 2008.*, Seiten 450–455, 2008.
- [ZDGH05] Olaf Zimmermann, Vadim Doubrovski, Jonas Grundler und Kerard Hogg. Service-Oriented Architecture and Business Process Choreography in an Order Management Scenario : Rationale , Concepts , Lessons Learned. In *OOPSLA '05 Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, Seiten 301–312, 2005.

