Klassifikation von Daten und Prozessen in Produktkonfiguratoren

Matthias Liebisch

Friedrich-Schiller-Universität Jena
Ernst-Abbe-Platz 2
07743 Jena
m.liebisch@uni-jena.de

ORISA Software GmbH
Humboldtstr. 13
07743 Jena
liebisch@orisa.de

Abstract: Der Einsatz sogenannter *Produktkonfiguratoren* ist mittlerweile für viele Unternehmen der Schlüssel im Wettbewerb um Marktanteile. Einerseits kann dadurch dem kundenseitigen Wunsch nach Individualisierung entsprochen und auf der anderen Seite dem wachsenden Kostendruck gerade im Zuge der Globalisierung begegnet werden. Neben den Gründen und Vorteilen, welche die Einführung eines Konfigurators für ein Unternehmen rechtfertigen, analysiert dieser Beitrag vor allem technische Aspekte. Dazu werden in einer Klassifikation verschiedene Kriterien der Datenhaltung im Kontext eines Konfigurators betrachtet, deren nachfolgende Bewertung zur Präsentation zweier klassischer Anwendungsszenarien führt. Darüber hinaus findet abschließend eine Beschreibung aktueller Herausforderungen für Produktkonfiguratoren statt.

1 Einleitung

In der aktuell durch Globalisierung und Individualisierung geprägten Welt stehen Firmen aller Branchen vor neuen Herausforderungen. Einerseits eröffnen sich durch den weltweiten Handel neue Märkte, welche jedoch vorwiegend einem harten Preiskampf unterliegen. Auf der anderen Seite wächst beim Kunden der Wunsch, möglichst individuelle Produkte zu erwerben, welche optimal seinen Bedürfnissen angepasst sind und ihm eine Abgrenzung zum massengefertigten „Produkt von der Stange“ erlauben. Zudem manifestiert sich durch die stetig wachsende Integration der Informationstechnologie in nahezu jedem Bereich eines Unternehmens der Wunsch, alle relevanten Prozesse lückenlos nachvollziehen und innerhalb der IT-Systeme abbilden zu können.

Für alle obigen Punkte stellt der sogenannte *(Produkt-)Konfigurator* eine Schlüsseltechnologie dar, zu dem nachfolgend grundlegende Begriffe und Einsatzmöglichkeiten erläutert werden. Abschnitt 2 nimmt daraufhin bezüglich der für einen Konfigurator relevanten Daten und Prozesse eine Klassifizierung vor. Für diese findet in Abschnitt 3 eine allgemeine Analyse und Bewertung statt, bevor abschließend Herausforderungen und offene Probleme im Abschnitt 4 vorgestellt werden sollen.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

1.1 Begrifflichkeiten

Produktkonfiguratoren sind vor allem in den produzierenden Wirtschaftszweigen, wie beispielsweise dem Automobil- und Maschinenbau[GS06], die Grundlage, dem Wunsch nach Individualisierung, sinkenden Produktionskosten und Beherrschung von fachlicher Komplexität gerecht zu werden. Erreicht wird dies durch eine konsequente *Modularisierung* der Produktpalette, wodurch eine gemeinsame Nutzung gleichartiger Baugruppen bzw. Bauteile in verschiedenen Endprodukten möglich ist. Unter Beachtung von zuvor gepflegten *Regeln* können dann im Konfigurator dem *Baukastenprinzip* folgend die verschiedenen Module zum gewünschten Produkt kombiniert werden. Dadurch wird sichergestellt, dass diese unter den Vorgaben von Entwicklung und Produktion auch tatsächlich gefertigt werden können.

Der sogenannte *Optionen-Konfigurator* ist dabei der am häufigsten verwendete Typ, welcher die möglichen Optionen für ein Produkt dem Nutzer präsentiert und die Individualisierung durch An- bzw. Abwahl stattfindet. Im Gegensatz dazu wird im *Zielgrößen-Konfigurator* das gewünschte Produkt mittels Festlegung gewisser Eigenschaften durch den Nutzer beschrieben, anschließend wählt der Konfigurator die zur Erfüllung der Anforderungen passenden Optionen. Beispielsweise wäre es denkbar, mit dem Wunsch „ Wähle das Modell mit dem sparsamsten Motor“ die Konfiguration eines Automobils zu starten, statt den Kunden vor die Auswahl einer Baureihe zu stellen.

1.2 Einsatzzwecke

Konfiguratoren haben sich über die letzten Jahre in den verschiedensten Branchen etabliert und lassen sich bezüglich Funktionalität, Aufgabenstellung und Einsatzgebiet in zwei Kategorien gruppieren. Einerseits bieten sie als Marketinginstrument im Endkundengeschäft (*B2C, Business-to-Consumer*) einen direkteren, aktuelleren und interaktiveren Produktzugang gegenüber klassischen Wegen wie Filialen oder Katalogen. Andererseits können sie die Prozesse von Entwicklung, Produktion und Vertrieb nahtlos verbinden, sowohl innerhalb eines Unternehmens als auch in Geschäftsbeziehungen zu anderen Unternehmen (*B2B, Business-to-Business*). Weitere Möglichkeiten für den Einsatz von Produktkonfiguratoren finden sich in [Kru10].

Aufgrund des Baukastenprinzips und einer sorgfältigen Modularisierung der Produktpalette können durch einen Konfigurator über Wiederverwendungs- und Synergieeffekte erhebliche Kosten gespart werden, beispielsweise durch Nutzung gleicher Bauteile in verschiedenen Modellen eines Automobil-Herstellers. In dieser Funktion helfen Konfiguratoren außerdem, die inhärente Komplexität zu beherrschen, indem der Nutzer in vielfältiger Weise beim Zusammenstellen seines gewünschten Produktes unterstützt wird[Sto07]. Hierzu zählen insbesondere die automatische Prüfung aller im System hinterlegten Regeln und Bedingungen nach jeder Änderung der Konfiguration sowie die Auflistung aller möglichen Optionen zu einem konkreten Modul, beispielsweise alle Felgentypen für ein gewähltes Fahrzeugmodell.

2 Klassifikation

Dieser Abschnitt beschreibt orthogonal zueinander stehende eher technische Klassifizierungsmerkmale und deren Ausprägungen bezüglich der Datenhaltung für Konfiguratoren. Dabei wird unterschieden nach der Semantik von Daten (Abschnitt 2.1), den Prozessen (Abschnitt 2.2), dem Architekturszenario (Abschnitt 2.3) und schließlich der Persistierungsschicht (Abschnitt 2.4), welche allgemein jedem Konfigurationssystem zu Grunde liegen. Eine Bewertung und Analyse von Abhängigkeiten zwischen den einzelnen Kriterien gibt Abschnitt 3. Die in [Kru10] präsentierte Globalklassifikation bietet dagegen eine Charakterisierung von Produktkonfiguratoren verstärkt aus Anwendungssicht mit größerem Detailgrad bezüglich der einzelnen Prozesse.

2.1 Datensemantik

Die Klassifikation der für einen Konfigurator relevanten Daten orientiert sich im vorliegenden Beitrag an der Differenzierung in konfigurationsbezogene, oberflächenspezifische, nutzerabhängige und administrative Daten, welche im Folgenden näher erläutert werden. Abhängig vom Integrationsgrad eines konkreten Konfigurators in die Unternehmensprozesse können möglicherweise weitere Datengruppen oder andere Unterscheidungsmerkmale existieren.

Modelldaten: Unter diesem Begriff sollen alle Daten verstanden werden, die das eigentliche Konfigurationswissen repräsentieren. Dies sind sowohl Informationen zu allen verfügbaren Modulen, Optionen und Regeln der zu konfigurierenden Produkte als auch die zugehörigen Beschreibungen wie Texte, Preise oder Bilder.

Oberflächendaten: In dieser Kategorie werden alle Informationen und Daten zusammengefasst, die unabhängig vom konkret zu konfigurierenden Produkt für die Präsentation des Konfigurators an sich benötigt werden. Hierzu zählen beispielsweise alle Texte und Bilder für die Oberflächenelemente wie Menüs und Buttons, aber auch Steuerdaten für die Layout-Definition. Diese Art von Daten werden in der folgenden Bewertung nicht explizit betrachtet, da sie für die Entwicklung der Konfigurator-Anwendung relevant sind, nicht aber für den eigentlichen Konfigurationsprozess.

Konfigurationsdaten: Diese Gruppe vereint alle in der Interaktion zwischen Anwender und Konfigurator relevanten Daten. Dazu zählen einerseits Informationen zu den vom Anwender gespeicherten Konfigurationen. Andererseits kann für weiterführende Prozesse, wie beispielsweise eine Angebotsgenerierung oder Bestellauslösung, die Aufnahme von Informationen zum Anwender bzw. Kunden (Anrede, Name, Anschrift) erforderlich sein, welche dann in geeigneter Art und Weise mit den eigentlichen Konfigurationsdaten verknüpft und persistiert werden müssen. Über Umfang und Detailgrad an Kundeninformationen entscheidet natürlich auch das Einsatzgebiet des Konfigurators, weil die Angabe von sensiblen privaten Informationen im Internet durchaus eine Hemmschwelle darstellt.

ISBN 978-3-88579-286-4

Administrationsdaten: Alle organisatorisch sowie technisch notwendigen Informationen für den Betrieb eines Konfigurators werden in dieser Gruppe zusammengefasst. Hierunter gliedern sich beispielsweise Daten für Nutzerberechtigungen, Rollenkonzepte und eine Sessionverwaltung ein.

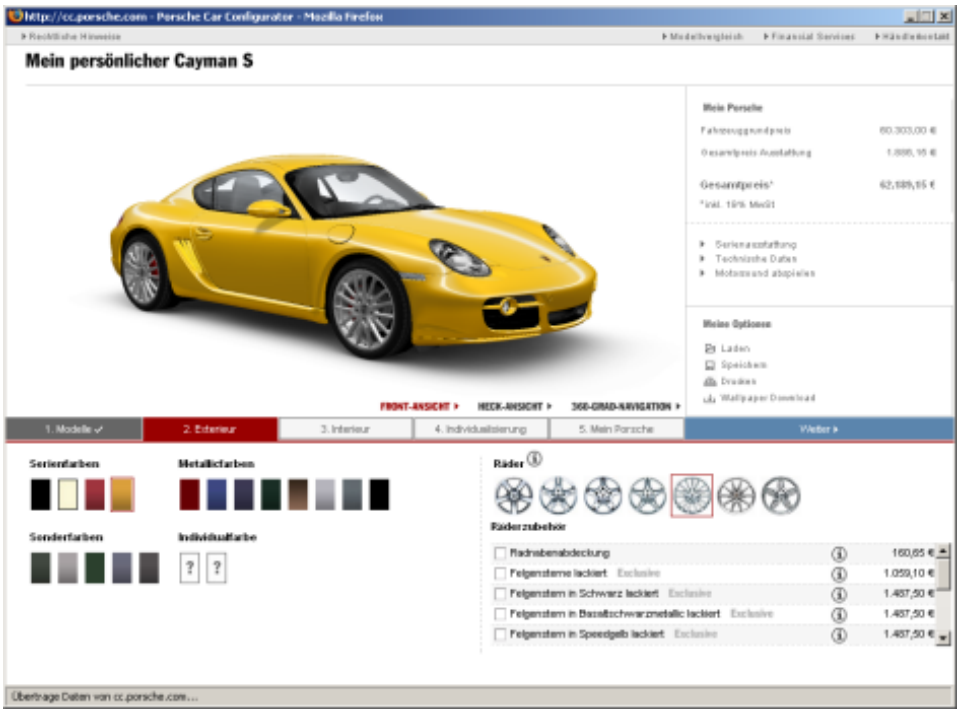

Abbildung 1: Beispiel eines Produktkonfigurators (Porsche Car Configurator)

Am Beispiel des im Internet verfügbaren *Porsche Car Configurator*[1] als eine von ORISA Software GmbH[2] entwickelte individuelle Konfigurations-Lösung im Umfeld der Automobilbranche soll die Trennung zwischen Modell-, Oberflächen- und Konfigurationsdaten verdeutlicht werden. Entsprechend der Darstellung in Abbildung 1, ist der Konfigurator in mehrere Bereiche unterteilt. Zu den modellspezifischen Daten gehören sowohl der gesamte untere Bereich mit den Außenfarbkacheln, Radkacheln und Räderzubehör-Optionen, die 3D-Visualisierungsdaten des Fahrzeugs in der oberen Hälfte als auch die Preisinformationen im rechten oberen Bereich.

Dagegen sind die restlichen Anzeigeelemente modellunabhängig und subsumieren somit die Oberflächendaten des eigentlichen Konfigurators. Hierzu zählt beispielsweise die Menüleiste, welche den oberen vom unteren Bereich trennt und die Auswahl der einzelnen Konfigurationsschritte (Modell, Exterieur, Interieur, Individualisierung, Übersicht)

[1] `http://www.porsche.com`
[2] `http://www.orisa.de`

ISBN 978-3-88579-286-4

ermöglicht. Weiterhin gehören in diese Gruppe die Buttons in der oberen Menüleiste (Rechtliche Hinweise, Modellvergleich, usw.) und die Funktionen im rechten Bereich zum Laden, Speichern und Drucken einer Konfiguration.

Um auf einen individuell zusammengestellten Porsche auch zu einem beliebigen späteren Zeitpunkt zugreifen zu können, muss der aktuelle Konfigurationsstand durch den Anwender gespeichert werden. Hierzu ist eine Registrierung inklusive Hinterlegung von persönlichen Informationen wie Name und E-Mail notwendig, um die Zuordnung zwischen Konfigurationen und Anwendern zu gewährleisten. Bei Bedarf findet die weitere Bearbeitung der im Internet erstellten Konfiguration bis zur finalen Bestellung in einem Porsche-Zentrum statt, in diesem Prozess werden die Konfigurationsdaten um weitere relevante Kundendaten ergänzt.

2.2 Workflow/Prozess

Die in Abschnitt 2.1 aufgeführten Daten unterliegen verschiedenen Prozessen im Kontext des Konfigurators und lassen sich in die anfängliche Modellierung, die Pflege und Bereitstellung der Produktinformationen sowie die Verarbeitung der Benutzerdaten unterscheiden. In Abbildung 2 ist das Zusammenspiel der einzelnen Schritte skizziert, welche nachfolgend näher erläutert werden. Allgemein bilden die Prozesse der Datenpflege und Datenbereitstellung einen iterativen Zyklus, der einerseits durch interne Anforderungen bzw. Anpassungen aus den Bereichen Produktion, Entwicklung und Marketing sowie andererseits auch von externen Kundenwünschen maßgeblich beeinflußt wird.

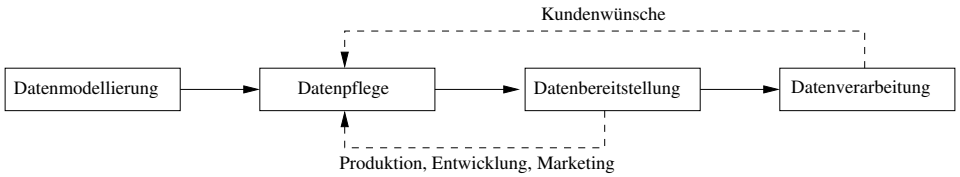

Abbildung 2: Grundlegende Prozesse im Umfeld eines Produktkonfigurators

Datenmodellierung: Um eine Produktpalette sinnvoll konfigurieren zu können, muss initial eine Analyse der Produktdaten sowie die Abbildung in die Welt des Konfigurators erfolgen. Dieser Schritt umfasst die sinnvolle Definition von Modulen im Sinne des Baukastenprinzips und sollte deswegen gewissenhaft und unter gemeinsamer Abstimmung aller an diesem Prozess beteiligten Unternehmensbereiche stattfinden, da spätere Modellanpassungen mitunter sehr aufwändig sein können.

Datenpflege: In einer zweiten Phase können die gewünschten Produkte entsprechend der Modellvorgaben modularisiert werden. Neben der Pflege von Optionen und der Hinterlegung von Semantik (z.B. Texte, Preise, Bilder) wie im Abschnitt 2.1 beschrieben, muss auch die Definition von Regeln und Abhängigkeiten erfolgen, welche die gewünschten Eigenschaften der zu konfigurierenden Produkte sicherstellen.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

Datenbereitstellung: Nach abgeschlossener Datenpflege kann der Konfigurator im einfachsten Fall direkt auf diese Daten zugreifen. Ist dies aus technischen oder organisatorischen Gründen nicht sinnvoll, beispielsweise um die Nutzungslast auf mehrere Systeme verteilen zu können, müssen die Daten aus dem Pflegesystem in ein oder mehrere Nutzsysteme transportiert werden. Dabei kommt es häufig zu einer Transformation der Daten und zu einem Wechsel der Persistierungsebene (siehe Abschnitt 2.4).

Datenverarbeitung: Die vom Nutzer getroffenen Auswahlentscheidungen können am Ende eines Konfigurationsprozesses in Form einer sogenannten *Konfiguration* gespeichert werden und stehen eventuell weiteren Verarbeitungsschritten, wie beispielsweise einer Angebotserstellung, zur Verfügung. Eine solche Konfiguration sollte es dem Nutzer ermöglichen, prinzipiell jederzeit durch Laden derselben den Konfigurator in den Zustand zum Zeitpunkt der letzten Speicherung zu versetzen. Dazu müssen neben den gewählten Optionen für ein Produkt auch Informationen zum Konfigurator selbst abgelegt werden, beispielsweise die zuletzt gezeigte Ansicht oder gewisse Selektionszustände.

2.3 Architekturszenario

Anwendungssysteme im Allgemeinen und Produktkonfiguratoren mit den in Abschnitt 2.2 aufgeführten Prozessen im Speziellen können abhängig vom gewünschten Anwendungsverhalten und den technischen Voraussetzungen in einem zentralen oder verteilten Szenario betrieben werden.

Zentral: Die Durchführung der verschiedenen Prozesse setzt in einem zentralen Client-Server-Szenario eine dauerhafte Verbundenheit zwischen Client und Server voraus, um die serverseitig persistierten Daten verarbeiten zu können. Zudem ist es in einer solchen Architektur möglich, die gesamte Anwendungs- und Konfigurationslogik ebenfalls an zentraler Stelle auf einem *Application Server* vorzuhalten und durch sogenannte *Thin Clients* verarbeiten zu lassen.

Dezentral: Durch Replikation von Daten ist es möglich, einen Teil der verschiedenen Prozesse ohne dauerhafte Verbindung zwischen Client und Server durchführen zu können. Somit kann beispielsweise die Konfigurationserstellung durch den Nutzer auf einem unverbundenen Client durchgeführt werden. Da beim Konfigurieren ausschließlich lesende Zugriffe auf das Konfigurationswissen stattfinden, sind Strategien zur Konfliktvermeidung bzw. Konfliktauflösung, wie sie üblicherweise in Szenarien teilweise unverbunden arbeitender Clients anzutreffen sind [Lie03], nicht notwendig. Einzig lokal gespeicherte Konfigurationen müssen gegebenenfalls mit dem Server synchronisiert werden können.

ISBN 978-3-88579-286-4

2.4 Persistierungsebene

Notwendige Voraussetzung für ein Konfigurationssystem ist die Persistierung der in Abschnitt 2.1 beschriebenen Daten. Hierfür bieten sich je nach Anforderung und Einsatzbereich des Produktkonfigurators verschiedene Möglichkeiten, die jedoch klassisch zwischen datenbanksystem- und dateibasiert unterschieden werden können. Sinnvoll ist mitunter auch ein Wechsel der Persistierungsebene zwischen den verschiedenen in Abschnitt 2.2 erläuterten Prozessen.

Datenbank: Relationale Datenbankmanagementsysteme (RDBMS) eignen sich vor allem aufgrund der Zusicherung der ACID-Eigenschaften zur Persistierung. Diese sind insbesondere für den ändernden Mehrbenutzerbetrieb im Prozess der Datenpflege relevant. Desweiteren können durch den standardisierten Zugriff per SQL die Resultate aus dem Konfigurationsprozess einfach genutzt werden in anderen Unternehmensbereichen, wie z.B. der Produktion. Die Möglichkeiten des relationalen Modells bzw. seiner Umsetzung in den entsprechenden Produkten sind allerdings nicht immer optimal geeignet, um die aus der Produktanalyse und der Datenmodellierung entstandenen Objekte wie Module, Optionen und Regeln in kanonischer Art und Weise auf die vorgegebenen relationalen Strukturen abzubilden. Der Einsatz objektrelationaler Systeme bzw. entsprechender Mapper ist einer der zukünftig zu analysierenden Themen, welche in Abschnitt 4 angesprochen werden.

Dateisystem: Die Speicherung von Daten in das vom Betriebssystem verwaltete Dateisystem ist unkompliziert und schnell, zudem fällt kein zusätzlicher Installations- und Wartungsaufwand an. Als Dateiformate bieten sich je nach Anwendungsgebiet die unterschiedlichsten Möglichkeiten, von einfachen ASCII-Dateien über XML-basierte Formate bis hin zu komplexen Binärformaten wie Excel-Dateien oder ZIP-Archiven. Problematisch sind dagegen Fragen der Konsistenzsicherung vor allem während der Datenpflege bei mehreren parallelen Nutzern sowie die Datensicherung. Weiterhin fehlt die implizite strukturelle Abbildung eines Datenmodells, weswegen es oft zur Verwendung proprietärer Datenstrukturen kommt.

3 Bewertung der Klassifikation

Basierend auf der in Abschnitt 2 vorgestellten Klassifikation soll in diesem Abschnitt eine Bewertung der einzelnen Kriterien vorgenommen werden. Diese folgt entsprechend dem in Abbildung 2 skizzierten Prozessablauf den Schritte zur Pflege, Bereitstellung und Verarbeitung der relevanten Daten eines Produktkonfigurators.

Sowohl für den initialen Prozess der **Datenmodellierung** als auch den iterativen Prozess der **Datenpflege** ist es aus organisatorischer Sicht sinnvoll, eine zentrale Persistierungslösung einzusetzen. Zudem stellt das Konfigurationswissen immer eine Synthese von Daten aus den Bereichen Produktion und Entwicklung dar, welche üblicherweise auch zentral gesteuerte Prozesse sind bzw. einer gegenseitigen Abstimmung unterliegen. Dabei

bieten Datenbanken bzw. Datenbankmanagementsysteme eine optimale Grundlage vor allem bei der Definition der Modelldaten, um diese konsistent, ausfallsicher und verfügbar für parallel arbeitende Nutzer persistieren zu können. Dies ist gerade in komplexen Konfiguratoren mit einer Vielzahl von Modulen, Optionen und Regeln relevant, da hier durchaus verschiedene Nutzergruppen auf überschneidenden Datenmengen arbeiten. Dennoch ist ein Szenario der verteilten Datenpflege mitunter anwendungsbedingt erforderlich und auch technisch realisierbar, beispielsweise für die länderspezifische Pflege von Modelldaten, welche nicht von einer zentralen Stelle durchgeführt werden soll oder kann. Diese Dezentralisierung wirft allerdings sofort Fragen bezüglich Replikation, Synchronisation und Versionsverwaltung zwischen den einzelnen Pflegesystemen auf und erhöht die Komplexität bei der Bereitstellung der Daten für den eigentlichen Konfigurationsprozess.

Nachdem die Konfigurations- und Administrationsdaten gepflegt sind, können diese entweder direkt vom Konfigurator verwendet werden oder müssen diesem im Prozess der **Datenbereitstellung** zur Verfügung gestellt werden. Häufig ist dieser Schritt auch mit einer Transformation der Daten sowie einem Systemwechsel verbunden, wodurch eine Optimierung bezüglich der Zielplattform sowie eine organisatorische bzw. sicherheitsrelevante Trennung zwischen Pflege und produktiver Verwendung von Modelldaten erreicht werden soll. Aufgrund der Tatsache, dass im eigentlichen Konfigurationsvorgang auf die benötigten Daten nur lesender Zugriff erfolgt, qualifiziert sich die dateisystembasierte Bereitstellung der Daten nach Extraktion aus der Pflegedatenbank in den meisten Fällen als vollkommen ausreichend. Dies gilt sowohl für die Nutzung zentraler Konfigurationsszenarien (siehe Abschnitt 3.1) als auch im besonderen Maße für Produktkonfiguratoren im dezentralen unverbundenen Einsatz (siehe Abschnitt 3.2), da bei Verwendung eines DBMS auf mobilen Endgeräten mit jedem Nutzer der Aufwand für Lizenzen, Wartung und Synchronisation steigt.

Schließlich steht im Prozess der **Datenverarbeitung** die Frage nach einer geeigneten Persistierung der vom Nutzer erstellten Konfigurationen. Neben den gewählten Modulen und Optionen zählen hierzu beispielsweise auch erfasste Kundendaten. Abhängig von Art und Anzahl der Anwender (Vertriebsmitarbeiter vs. Internetnutzer), dem Prozessziel von erstellten Konfigurationen (Grundlage zur Angebotserstellung vs. Definition von Produktionsstücklisten) sowie dem Architekturszenario (zentral vs. unverbunden) kommen sowohl Datenbanken als auch das Dateisystem als Persistierungsebene zum Einsatz. Eine hohe Anzahl parallel arbeitender Nutzerzahl inklusive der schreibenden Zugriffe beim Speichern von Konfigurationen wird im zentralen Client-Server-Szenario durch eine Datenbank optimal unterstützt. Diese bietet zugleich eine integrierbare Ausgangslage für eventuell nachfolgende Prozesse. Allerdings muss eventuell ein passendes Berechtigungssystem dafür sorgen, dass trotz zentraler Ablage Nutzer nur die eigenen Konfigurationen lesen und bearbeiten können. Wird stattdessen eine lokale Persistierung angestrebt, ist häufig das Dateisystem aus den im letzten Absatz genannten Gründen ausreichend. Unabhängig davon muss jedoch bei einer Datenspeicherung auf unverbunden konfigurierenden Clients dafür gesorgt werden, dass eine Synchronisation und Sicherung der lokal erzeugten Nutzerdaten serverseitig stattfindet. Dies ist vor allem bei einer prozessbedingten Weiterverarbeitung von Konfigurationen und Konsolidierung von Kundendaten durchaus geschäftskritisch.

ISBN 978-3-88579-286-4

	zentral		dezentral	
	Datenbank	Dateisystem	Datenbank	Dateisystem
Datenmodellierung	++	–	––	––
Datenpflege	++	–	––	––
Datenbereitstellung	+	++	–	++
Datenverarbeitung	++	––	–	++

++ : gut geeignet + : geeignet – : bedingt geeignet –– : ungeeignet

Tabelle 1: Bewertung von Kombinationsmöglichkeiten verschiedener Klassifikatoren

In Tabelle 1 ist die vorangegangene Bewertung zusammengefasst dargestellt. Hierbei werden die unterschiedlichen Kombinationsmöglichkeiten der Kategorien Prozess, Architekturszenario und Persistierungsebene über vier Abstufungen von *gut geeignet* (++) bis *ungeeignet* (––) differenziert. Basierend auf diesen Ergebnissen sollen in den nächsten Abschnitten zwei praxisrelevante Szenarien skizziert und erläutert werden.

3.1 Zentrales Konfigurationsszenario als Webapplikation

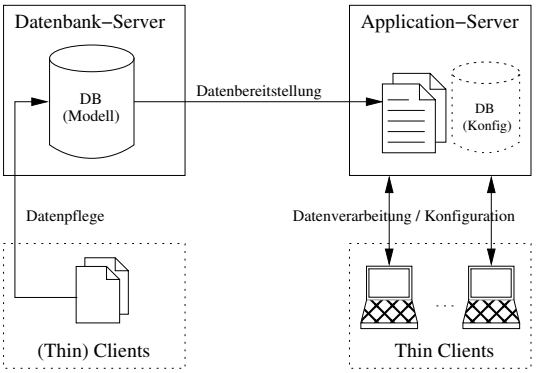

Abbildung 3: Zentrales Konfigurationsszenario mit dauerhaft verbundenen Clients

Ein typisches zentrales webbasiertes Konfigurationsszenario, wie in Abbildung 3 dargestellt, ist durch folgende Eigenschaften charakterisiert. Die Modelldaten werden in einer zentralen Datenbank gepflegt und anschließend durch einen Transformationsprozess in passenden Strukturen als Dateien einem zentralen *Application-Server* bereitgestellt, auf dem der Konfigurator als Web-Anwendung betrieben wird. Durch die dauerhaft verbundenen Clients können die verarbeiteten Konfigurationsdaten wiederum in einer zentralen Datenbank persistiert werden, welche im Bild durch das gestrichelte Objekt mit der Beschriftung „DB (Konfig)“ markiert ist. Aus dieser ist dann die Versorgung nachgelagerter Prozesse, wie z.B. eines Bestellsystems, über entsprechende Schnittstellen möglich.

Da sowohl die Daten als auch die gesamte Anwendungslogik auf dem Application-Server vorgehalten werden, ist einerseits die Pflege und Wartung der gesamten Anwendung an zentraler Stelle einfach möglich. Andererseits genügen zur Verwendung des Konfigurators sogenannte *Thin Clients*, in diesem Fall einfache Web-Browser. Deswegen ist diese Architektur optimal geeignet für die Bereitstellung eines Produktkonfigurators sowohl zum „Erkunden“ einer Produktpalette für Internetnutzer als auch zur Unterstützung von Verkäufern und Händlern im direkten Kundenprozess innerhalb eines Intranets.

3.2 Dezentrales Konfigurationsszenario

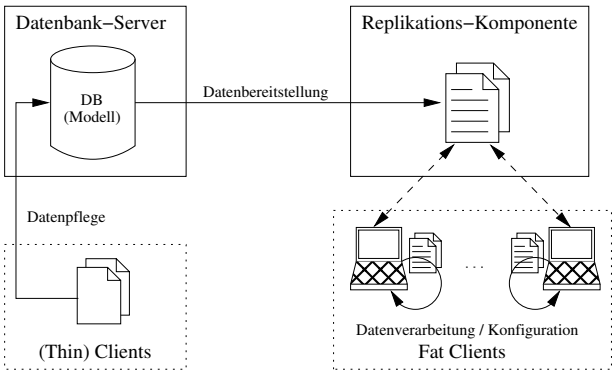

Abbildung 4: Dezentrales Konfigurationsszenario mit mobilen Clients

Wie in Tabelle 1 verdeutlicht, sollte das Konfigurationswissen möglichst zentralisiert erfasst werden. Deshalb basieren auch in einem dezentralen Konfigurationsszenario (siehe Abbildung 4) die Prozesse der Datenmodellierung und Datenpflege auf einer zentralen Datenbank. Der im Anschluß folgende Prozess der Datenbereitstellung berücksichtigt jedoch bereits, dass ein Konfigurieren auf mobilen Clients ohne dauerhafte Server-Verbindung ermöglicht werden soll, beispielsweise für Präsentations- und Planungszwecke beim Kunden. Eine *Replikationskomponente* sorgt dafür, dass jedem verbundenen Client der aktuelle Datenbestand bereitgestellt wird. Bedingt durch die anschließende Unverbundenheit können Ergebnisse der Datenverarbeitung vorerst nur lokal persistiert werden, beispielsweise im Dateisystem des Clients.

Neben den eigentlichen Modelldaten muss auf jedem Client zudem die gesamte Anwendungslogik des Konfigurators vorhanden sein, welche ebenfalls bei Änderungen aktualisiert werden muss. Dieser erhöhte Wartungsaufwand ist die Konsequenz für die Unterstützung des lokalen Konfigurierens auf sogenannten *Fat Clients* gegenüber dem in Abschnitt 3.1 beschriebene zentralen Szenario. Sollen die daraus resultierenden Ergebnisse in weiterführenden Prozessen verarbeitet werden, beispielsweise als Materialstückliste in der Produktion, müssen die lokal gespeicherten Konfiguration beim nächsten Synchronisationsvorgang durch die *Replikationskomponente* zum Server übertragen werden.

4 Herausforderungen

Nachdem in den vorangegangenen Abschnitten die Datenhaltung und verbundene Prozesse in Konfiguratoren vorgestellt und klassifiziert wurden, sollen hier einige Herausforderungen verdeutlicht werden, welche im Umfeld eines Konfigurators in der Praxis auftreten.

4.1 Referenzarchitektur

Seit dem ersten Auftreten vor über zwei Jahrzehnten sind Produktkonfiguratoren mittlerweile in den verschiedensten Wirtschafts-Branchen anzutreffen und haben sich zu hoch integrierten Anwendungssystemen entwickelt, welche eine optimale Verbindung der einzelnen Bereiche eines Unternehmens ermöglichen. Allerdings existieren bisher weder eine Referenzarchitektur für Produktkonfiguratoren noch eine Norm für die relevanten Modelldaten, wie dies beispielsweise im Kontext von (relationalen) Datenbankmanagementsystemen der Fall ist mit dem ANSI/SPARC-Datenmodell[Jar77] und SQL[ISO08] als Anfragesprache.

Diese Entwicklung hat dazu geführt, dass die einzelnen am Markt agierenden Software-Hersteller für Konfigurationssysteme[RP03, GS06] jeweils eine eigene Lösung bezüglich der bereits in Abschnitt 2 aufgeführten Kriterien anbieten. Unter diesen Voraussetzungen versucht KRUG mit [Kru10] aufzuzeigen, welche grundlegenden Prozesse und Schnittstellen in einem Produktkonfigurator unabhängig vom Anwendungskontext existieren sollten und wie ein Referenzmodell hierfür aussehen könnte. Dies sind Voraussetzungen, um eine Kompatibilität der Systeme analog zu Produkten für RDBMS zu ermöglichen und damit eine größere Herstellerunabhängigkeit zu gewährleisten.

4.2 Aspektorientierte Datenhaltung

Gerade durch die in Abschnitt 1 skizzierten Einsatzgründe für Produktkonfiguratoren, besteht meist neben der rein fachlichen Abbildung des Baukastens als Modelldaten die Anforderung, zusätzliche und applikationsweite Funktionalität zu unterstützen. Hierzu gehören klassischerweise die internationalisierte[Wes92, HBL02, Göb09] und versionierte Darstellung sowie Speicherung der im Konfigurator verwendeten Daten, insbesondere also die datenbankgestützte Pflege und Verwaltung landesspezifischer Texte, Preise und Währungsformate von Modulen und Optionen. Aus rein technischer Sicht erfüllen mittlerweile alle namhaften DBMS-Produkte diesen Anspruch durch Unterstützung von Unicode, wobei noch einzelne Probleme im Rahmen der Integration[KH03] existieren. Jedoch fehlen aktuell noch Strategien, um derartige Anforderungen orthogonal zur Definition eines konkreten Datenmodells unterstützen zu können.

Die Aspektorientierte Datenhaltung[Lie10a] definiert ein Modellierungs-Paradigma, um die beschriebenen querschnittlichen Funktionen bzw. funktionalen Aspekte in einem gegebenen Datenmodell möglichst modular und mit geringen Anpassungen integrieren zu

können. Für das relationale Datenmodell basiert diese Erweiterung auf einer festen Menge von Verwaltungstabellen als Infrastruktur[Lie10b], welche in einem ersten einem konkreten Datenmodell hinzugefügt werden müssen. Im zweiten Schritt können nun zu den Originaldaten aspektspezifische Werte gepflegt werden. Der Zugriff und die Auswertung dieser zusätzlichen Daten sollte aus Gründen der Performance[LP10] und Bedienbarkeit über eine entsprechende Schnittstelle[Pie11] vorgenommen werden.

4.3 Datenmodelle zur Persistierung

Grundlage für die Datenhaltung in Konfiguratoren ist eine geeignete Form der Speicherung aller relevanten Daten über die verschiedenen Prozesse hinweg. Neben der vom Anwendungskontext abhängigen Form der Datenbereitstellung bzw. Datenverarbeitung im eigentlichen Konfigurationsprozess hat die Diskussion in Abschnitt 3 gezeigt, dass die Datenpflege des Konfigurationswissens zentraler datenbankbasierter Prozess etabliert werden sollte. Hierfür werden aus Gründen der Marktdurchdringung und Praxisrelevanz vor allem relationale Datenbanksysteme eingesetzt, welche jedoch bezüglich zu speichernden Objekte und Anforderungen nicht immer eine optimale Grundlage bilden. Deswegen sollten alternative Persistierungsformen, beispielsweise objektrelationale bzw. objektorientierte Datenbanksysteme[HSS07, MW03] oder XML[HM04], analysiert und bezüglich verschiedener Kriterien wie Unterstützung paralleler Nutzer, Transaktionshandling, Performance, Abfragesprachen und Unterstützung funktionaler Aspekte verglichen werden.

Literatur

[Göb09] Andreas Göbel. Internationalisierung in Produktkonfiguratoren - Anforderungen und Konzepte für die Datenhaltung. Diplomarbeit, Fakultät für Mathematik und Informatik, Friedrich-Schiller-Universität Jena, Marz 2009.

[GS06] Norbert Gronau und Simone Schmid. Marktüberblick: Konfiguratoren in ERP-/PPS-Systemen. *PPS Management*, 10(1):55–61, 2006.

[HBL02] Z. He, D. W. Bustard und X. Liu. Software internationalisation and localisation: practice and evolution. In *PPPJ '02: Proceedings of the inaugural conference on the Principles and Practice of programming, 2002*, Seiten 89–94, Maynooth, County Kildare, Ireland, 2002. National University of Ireland.

[HM04] E. Rusty Harold und W. Scott Means. *XML in a Nutshell: A Desktop Quick Reference*. O'Reilly Media, 2004.

[HSS07] Andreas Heuer, Gunter Saake und Kai-Uwe Sattler. *Datenbanken. Konzepte und Sprachen*. Mitp-Verlag, 2007.

[ISO08] *ISO/IEC 9075-2:2008. Information technology – Database languages – SQL – Part 2: Foundation (SQL/Foundation)*. ISO, Geneva, Switzerland, 2008.

[Jar77] Donald A. Jardine. *ANSI/SPARC DBMS Model*. Elsevier Science Publishing Co Inc., 1977.

[KH03] A. Kumaran und Jayant R. Haritsa. On the cost of multilingualism in database systems. In *VLDB '03: Proceedings of the 29th International Conference on Very Large Data Bases*, Seiten 105–116. VLDB Endowment, 2003.

[Kru10] Andreas Krug. Entwurf eines integrativen Grundmodells für Produktkonfiguratoren. Diplomarbeit, Institut für Informatik, Friedrich-Schiller-Universität Jena, Juni 2010.

[Lie03] Matthias Liebisch. Synchronisationskonflikte beim mobilen Datenbankzugriff: Vermeidung, Erkennung und Behandlung. Diplomarbeit, Institut für Informatik, Friedrich-Schiller-Universität Jena, Februar 2003.

[Lie10a] Matthias Liebisch. Aspektorientierte Datenhaltung - ein Modellierungsparadigma. In *Proceedings of the 22nd GI Workshop Grundlagen von Datenbanken 2010*, Seiten 13–17, Bad Helmstedt, Mai 2010.

[Lie10b] Matthias Liebisch. Supporting functional aspects in relational databases. In *Proceedings of the 2nd International Conference on Software Technology and Engineering (ICSTE)*, Jgg. 1, Seiten 227–231, San Juan, Puerto Rico, Oktober 2010.

[LP10] Matthias Liebisch und Matthias Plietz. Performance-Analysen für Realisierungsansätze im Kontext der Aspektorientierten Datenhaltung. Institut für Informatik, Friedrich-Schiller-Universität Jena, November 2010.

[MW03] Andreas Meier und Thomas Wüst. *Objektorientierte und objektrelationale Datenbanken. Ein Kompass für die Praxis*. Dpunkt Verlag, 2003.

[Pie11] Bernhard Pietsch. Entwurf einer Zugriffsschicht für funktionale Aspekte in DBMS. Studienarbeit, Institut für Informatik, Friedrich-Schiller-Universität Jena, Marz 2011.

[RP03] Timm Rogoll und Frank Piller. *Konfigurationssysteme für Mass Customization und Variantenproduktion (Marktstudie)*. ThinkConsult, 2003.

[Sto07] Henrik Stormer. Kundenbasierte Produktkonfiguration. *Informatik-Spektrum*, 30(5):322–326, 2007.

[Wes92] G. D. A. (David) Weston. National language architecture. In *CASCON '92: Proceedings of the 1992 conference of the Centre for Advanced Studies on Collaborative research*, Seiten 59–69. IBM Press, 1992.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html