

K-Modell 1.3 - Strukturierter Entwurf von Konfigurationsmodellen

Dr. Axel Brinkop

Mit der Methode K-Modell wurde der Notwendigkeit Rechnung getragen, produktspezifische Konfigurationsmodelle systematisch zu entwickeln. Die Methode besteht aus zwei Komponenten: Einer Beschreibungsmethode für Konfigurationswissen und einer Methode zur Erarbeitung des Wissens. K-Modell geht von einer Trennung von Strukturwissen, Wissen über Abhängigkeiten und zugrunde liegenden Produktdaten aus, um eine effiziente Entwicklung und Pflege des Wissens zu gewährleisten. K-Modell hat sich in einer Vielzahl von Projekten in der Praxis bewährt.

Beschriebene K-Modell Version: 1.3.1

1. Die Methode

In der industriellen Praxis versteht man unter einem Produktkonfigurator ein Werkzeug, das hilft, ein Produkt nach vorgegebenen Anforderungen zu spezifizieren. In der Forschung unterscheidet man differenzierter zwischen dem Lösen einer Konfigurationsaufgabe, bei der aus einer vorgegebenen Menge von Objekten ausgewählt wird und Parametrierung, bei der die Eigenschaften des gesuchten (Teil-)Objekts frei spezifiziert werden (vgl. [Brinkop 1999]). Mit K-Modell können beide Arten von Aufgaben beschrieben werden.

Softwaretechnisch wird ein werkzeuggestützter Produktkonfigurator dadurch realisiert, dass durch eine domänenunabhängige Konfigurationssoftware ein für eine Produktfamilie spezifisches Konfigurationsmodell ausgewertet wird. Das Konfigurationsmodell beinhaltet das Konfigurationswissen.

K-Modell ist eine softwareunabhängige Methode sowohl zur Beschreibung als auch zur Erarbeitung von Konfigurationsmodellen. Sie wurde in [Brinkop 2009] zum ersten Mal unter diesem Namen vorgestellt und seitdem kontinuierlich weiterentwickelt. Die Methode basiert auf der Annahme, dass Strukturwissen, Wissen über Abhängigkeiten und zugrunde liegende Daten getrennt repräsentiert werden (vgl. auch [Brinkop 1999; S. 90ff]). Abbildung 1 stellt die Wissensarten in ihrer Rolle im Konfigurationsprozess dar. Im Konfigurationsprozess werden durch das Konfigurationswissen die geeigneten Objekte aus der Grundmenge ausgewählt und zu einer Konfiguration mit den gewünschten Eigenschaften kombiniert.

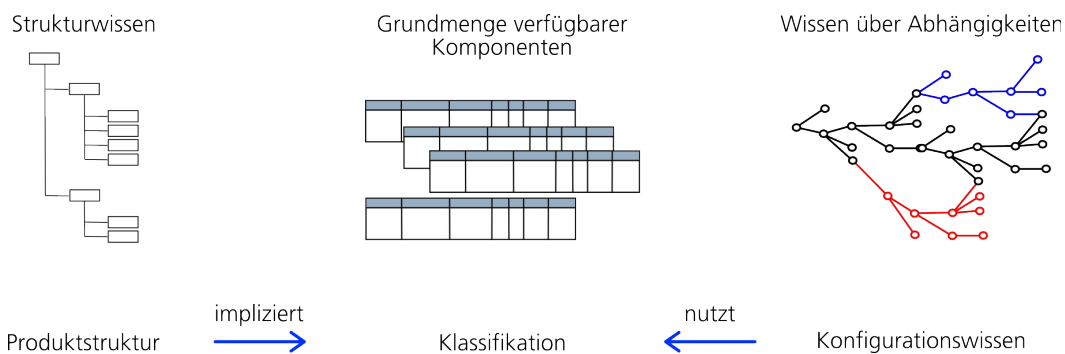


Abbildung 1: Wissensarten

Der Schwerpunkt der K-Modell-Methode liegt in Aufgabenstellungen, die durch Auswahl aus einer vorgegebenen Grundmenge von Komponenten gelöst werden können. Aufgaben der Parametrierung

können ebenso damit formuliert werden.

In der Abbildung 1 rechts ist das Wissen über Abhängigkeiten dargestellt. Durch dieses Wissen werden aus den zugrunde liegenden klassifizierten Produktdaten - in der Mitte dargestellt - die geeigneten Komponenten ausgewählt, entweder automatisch oder interaktiv mit dem Nutzer. Auf diese Weise können sowohl Aufgaben der Konfiguration und der Parametrierung als auch der Selektion und Suche gelöst werden.

Die zugrunde liegenden Komponenten (Baugruppen, Artikel, ...) werden in der Klassifikation gepflegt. Gleichartige Komponenten werden aufgrund gleichartiger Merkmale zu Klassen gruppiert, jede Komponente ist durch die konkreten Werte (Ausprägungen) der Merkmale beschrieben.

Im Konfigurationswissen ist die Struktur der Konfigurationsobjekte, die Abhängigkeiten zwischen den Komponenten und Methoden zur Bestimmung von Konfigurationsgrößen hinterlegt.

Durch die unterliegende Konfigurationsengine wird das Konfigurationswissen mit den Nutzereingaben ausgewertet, um die Anforderungen an die Komponenten zu ermitteln.

K-Modell geht von der Trennung von Konfigurationsmodell und Konfigurationsengine aus. Im Konfigurationsmodell werden keine Methoden zur Auswertung der Abhängigkeiten formuliert, K-Modell geht davon aus, dass die unterliegende Konfigurationsengine dazu in der Lage ist. Dabei dient kein spezielles Konfigurationswerkzeug als Vorbild. Die Anforderungen sind softwareneutral und können durch verschiedene am Markt erhältliche Konfigurationswerkzeuge erfüllt werden.

K-Modell ist aus einer Vielzahl von Projekten entstanden. Die Methode wurde für Produktexperten entwickelt, IT-Vorkenntnisse sind nicht notwendig. Sie bildet die Grundlage zur abteilungsübergreifenden Kommunikation über das Vorgehen zur Lösung von Konfigurationsaufgaben.

Die Beschreibungsmethode basiert auf der Darstellung der Zusammenhänge in Form einer Mindmap mit vorgegebener Struktur und fest vorgegebenen Schlüsselwörtern. Die Erstellung der Mindmap erfolgt mit dem OpenSource-Werkzeug Freeplane [Freeplane 2014], dessen Bedienung sehr schnell und einfach zu erlernen ist.

Die Erfahrung zeigt, dass die Methode sehr gut geeignet ist, in abteilungsübergreifenden Workshops mit Teilnehmern aus F&E, Produkt-Management und Vertrieb Konfigurationsmodelle interaktiv zu entwickeln. Durch das Arbeiten mit bekannten Produktzusammenhängen wird der Formalismus von den Workshop-Teilnehmern sehr schnell verstanden und die Diskussionen konzentrieren sich auf das produktspezifische Vorgehen zur Konfiguration. Die Inhalte sind für Fachexperten ohne IT-Vorkenntnisse gut verständlich, auch „Nicht-Techniker“ nehmen schnell an der Diskussion teil. Auf dieser Basis können diese abgestimmt definiert und zur Umsetzung in ein Software-Werkzeug freigegeben werden.

2. Das Beispiel

Als durchgängiges Beispiel dient die Konfiguration eines Designtisches. Den Designtisch gibt es in verschiedenen Grundformen: als Schreibtisch mit optionalem Seitenteil, als Besprechungstisch, als Stehtisch oder als Couchtisch. In Abbildung 2 ist er als Besprechungstisch dargestellt.



Abbildung 2: Besprechungstisch

Der Tisch besteht aus einer Tischplatte und einem Tischgestell. Das Gestell gibt es in verschiedenen Abmessungen, je nach Grundform in verschiedenen Grenzen. Je nach Grundform ist die Versteifung als „X“, „U“ oder „I“ ausgeführt.

Die Tischplatte gibt es aus Glas oder Holz in individuellen Abmessungen.

Sowohl Tischplatten als auch Gestelle werden kundenindividuell gefertigt, es gibt jedoch einige Gestelle, die auf Lager liegen.

3. Die Grundmenge

Die Grundmenge der zur Verfügung stehenden Komponenten wird in einer Klassenhierarchie mit Angabe der klassifizierenden Merkmale spezifiziert. Die Hierarchie wird durch das Schlüsselwort **CLASSES** gekennzeichnet.

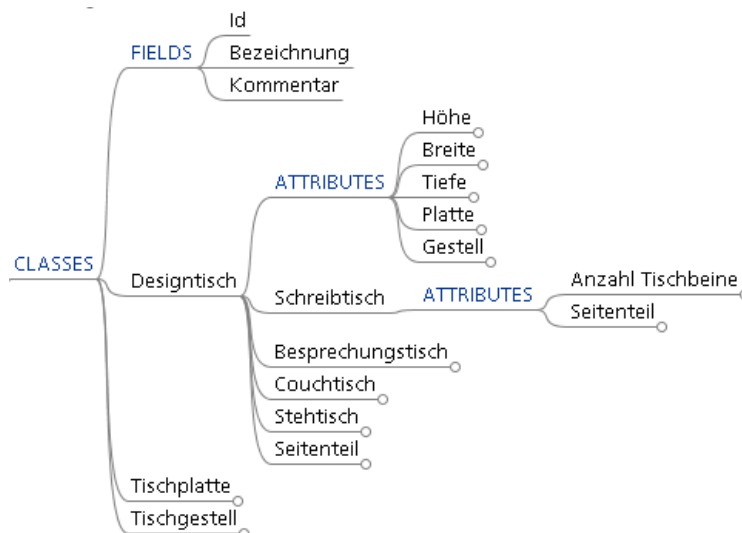


Abbildung 3: **CLASSES** - zugrunde liegende Klassen

Unter dem Schlüsselwort **ATTRIBUTES** werden die klassifizierenden Merkmale aufgelistet, wobei für jedes Attribut zusätzlich der Datentyp angegeben werden muss (**STRING**, **INT**, **NUM**, **BOOL**, **OBJECT**) und bei Aufzählungstypen die möglichen Werte nach dem Schlüsselwort **ONE-OF**.

Untergeordnete Klassen werden ohne weiteres Schlüsselwort aufgeführt. Attribute werden entlang der Hierarchie vererbt, in der untergeordneten Klasse sind alle Attribute der übergeordneten Klasse ebenfalls definiert. Auf jeder Hierarchieebene können Attribute hinzukommen, jedoch niemals wegfallen.

Mit einem K-Modell Add-on für Freeplane kann Vererbung simuliert werden, die vererbten Merkmale werden mit ihren Merkmalen unter dem Schlüsselwort **INHERITED** aufgelistet.

Die Strukturierung in Unterkomponenten lässt sich über Attribute des Datentyps **OBJECT** realisieren. In Abbildung 4 ist ein Beispiel zur Strukturierung des Tisches in Tischplatte und Tischgestell wiedergegeben.

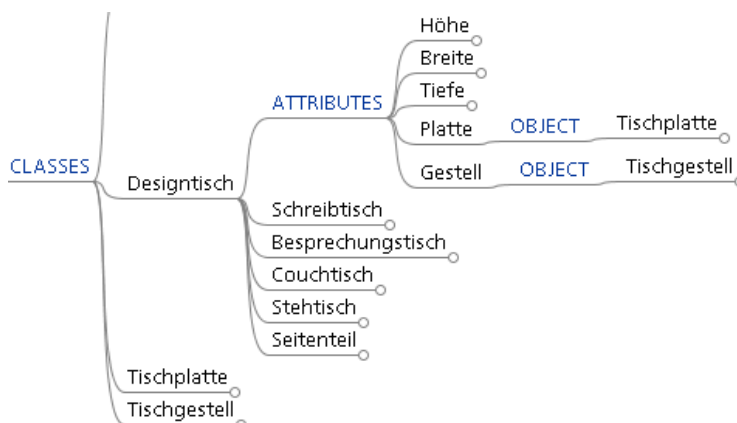


Abbildung 4: Unterstrukturierung des Tisches

Weiterhin wird zwischen primären und sekundären Merkmalen unterschieden. Sekundäre Merkmale lassen sich aus den anderen Merkmalen ohne zusätzliche Informationen herleiten. Beispielsweise lässt sich die Fläche der Tischplatte aus der Breite und Tiefe der Platte berechnen. Sekundäre Merkmale werden dadurch gekennzeichnet, dass sie eine Berechnungsvorschrift haben. Berechnungsvorschriften werden im K-Modell einheitlich nach dem gleichen Muster formuliert. Sie werden in Abschnitt 4.1 vorgestellt.

Durch die Spezifikation im Mindmap werden die Klassen definiert. Die Instanzen der Klassen werden unter dem Schlüsselwort INSTANCES den Klassen zugeordnet. Sie können dort direkt tabellarisch aufgelistet werden oder davon getrennt in externen Tabellen spezifiziert werden. Externe Tabellen können in Form von Excel-Arbeitsblättern, Datenbank-Tabellen oder CSV-Dateien vorliegen. Die Form hängt von den speziellen Vorlieben der Workshop-Teilnehmer und IT-Rahmenbedingungen ab und können auch später gewechselt werden.

Die Definition der Tabelle im K-Modell wird durch das K-Modell Community Add-on unterstützt, indem eine leere Tabelle mit allen zu definierenden Attributen als Spalten angelegt wird. Dabei werden Stammdaten-Felder und vererbte Attribute berücksichtigt. Ein Austausch der tabellarischen Daten mit Excel ist mit den Standard-Windows-Funktionalitäten möglich. Abbildung 5 zeigt einen Ausschnitt aus den verfügbaren Komponenten für das Beispiel integriert in das Mindmap.

id	bezeichnung	kommentar	hoehe	breite	tiefe	versteifung	beinzahl	vk2
schreib.001	Schreibtischgestell		700	1700	700	u	4	500,00
seite.001	Seitenteilgestell		700	1150	500	i	2	400,00
besprech.001	Besprechungstischgestell		700	1500	700	x	4	450,00

Abbildung 5: Tabelle der verfügbaren Komponenten

4. Die Konfiguration

Das Schlüsselwort CONFIGURATORS leitet das Thema „Konfiguration“ ein. Darunter können mehrere Konfiguratoren definiert werden, die sich auch gegenseitig referenzieren können. Damit lassen sich auch hierarchische Konfigurationsmodelle definieren. Die Konfigurationsmodelle müssen nicht alle in einer Mindmap definiert sein, sie können auch in anderen, externen Mindmaps definiert sein. Durch die Möglichkeit der Verlinkung von Mindmaps lässt sich bequem navigieren und man behält den Überblick.

Zu jedem Konfigurator werden jeweils Fragen und Ergebnisse formuliert.

4.1 Die Fragen

Konfigurationsgrößen werden im K-Modell durch Variablen beschrieben. Sie repräsentieren die Anforderungen an das zu konfigurierende Objekt und werden durch das Schlüsselwort QUESTIONS eingeleitet. Thematisch zusammengehörende Variablen werden zu sogenannten „Themen“ zusammengefasst. Themen können beliebig tief geschachtelt werden. In Abbildung 6 ist ein Ausschnitt aus den Variablen zur Konfiguration des Beispiels wiedergegeben. Sie sind in die Themen Tisch, Platte und Gestell strukturiert.

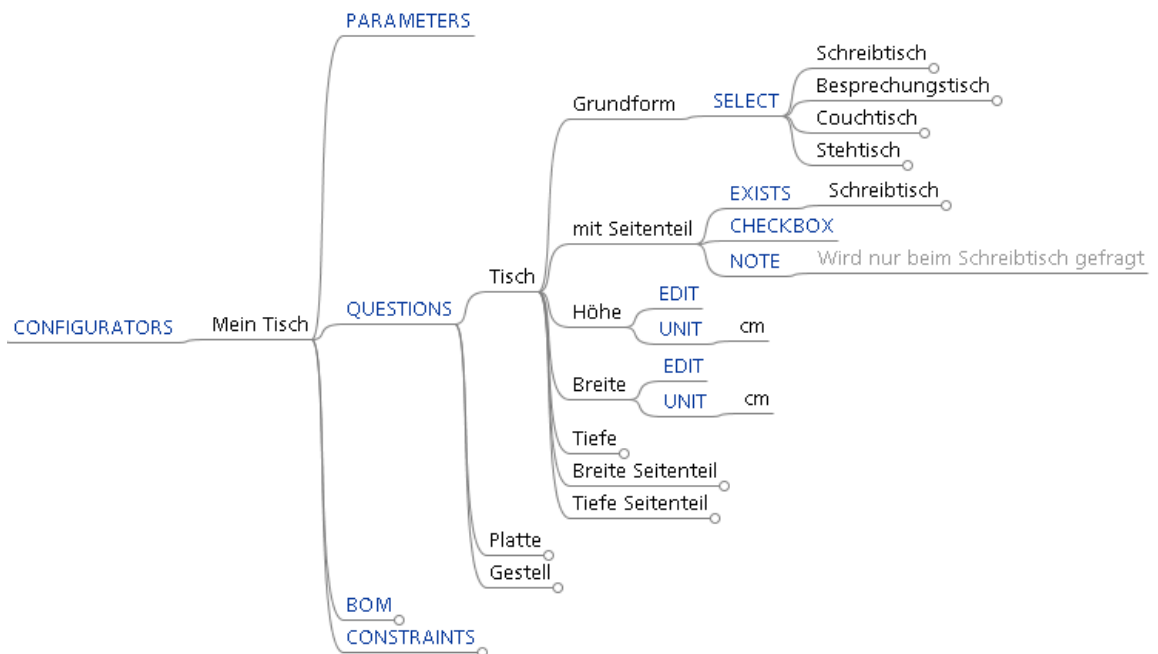


Abbildung 6: QUESTIONS - Konfigurationsgrößen

Zur hierarchischen Configuration kann ein Thema auch auf ein anderes Konfigurationsmodell verweisen. Dann darf es jedoch nicht zusätzlich Fragen oder Unterthemen enthalten.

Im Laufe des Konfigurationsprozesses werden die Variablen entweder interaktiv durch den Nutzer oder durch die Konfigurationsengine bestimmt, um dann zur Auswahl der Komponenten genutzt werden zu können.

Im K-Modell wird vereinfachend davon ausgegangen, dass es keine zusätzliche Ebene zur Definition der Benutzer-Schnittstelle gibt, sondern dass die Variablen selbst an der Oberfläche repräsentiert sind. Um auszudrücken, dass eine Variable interaktiv durch den Benutzer bestimmt werden soll, ordnet man der Größe oberflächenrelevante Eigenschaften zu. Zur Verfügung stehen `SELECT`, `EDIT` und `CHECKBOX` für eine Auswahlliste, ein Editierfeld und ein Markierungsfeld mit zwei Zuständen.

In Abbildung 7 finden Sie Beispiele für die Spezifikation der Oberflächeneigenschaften von Konfigurationsgrößen.

K-Modell geht von einem sprachunabhängigen Konfigurationsmodell aus. Für die Konfigurationsgrößen sind sprachneutrale Namen mittels dem Schlüsselwort `CODE` spezifiziert. Zusätzlich werden für die Konfigurationsgrößen sprachabhängige Übersetzungen angegeben. Für Deutsch findet man sie unter dem Schlüssel `LANG: de`, für Englisch unter dem Schlüssel `LANG: en`, analog für andere Sprachen.

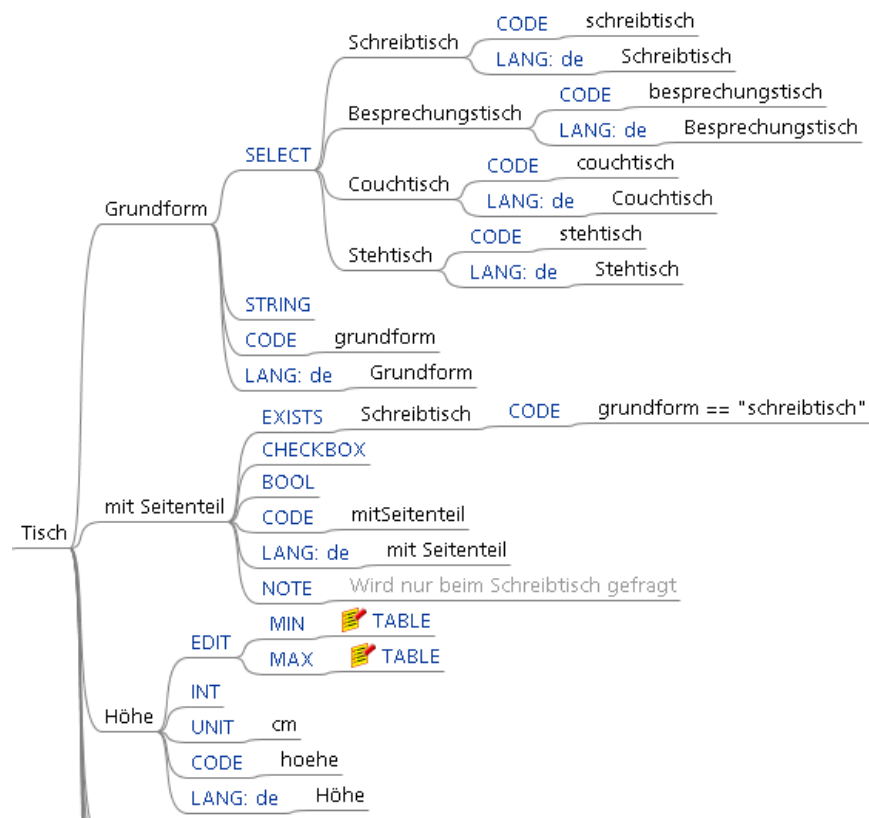


Abbildung 7: Details der Spezifikation von Konfigurationsgrößen

Die Umsetzung in eine konkrete Oberfläche hängt von der maximalen Schachtelungstiefe der Themen ab. Einigt man sich auf eine maximale Schachtelungstiefe von 2 Ebenen, so kann die Strukturierung mittels der Oberflächenelemente Register und Registerblätter umgesetzt werden. Abbildung 8 zeigt eine Skizze für eine Umsetzung des Tisch-Beispiels mit Registerblättern.

The figure shows three sequential screenshots of a configuration interface for a table, each with tabs for 'Tisch', 'Platte', and 'Gestell'.

First Screenshot (Tisch tab):

- Grundform: Dropdown menu
- mit Seitenteil: Checkbox
- Höhe: Input field
- Breite: Input field
- Tiefe: Input field
- Breite Seitenteil: Input field
- Tiefe Seitenteil: Input field

Second Screenshot (Platte tab):

- Plattenmaterial: Dropdown menu
- Plattenfarbe: Dropdown menu
- Plattenstärke: Dropdown menu

Third Screenshot (Gestell tab):

- Gestellversteifung: Input field
- Gestell auf Lager: Input field

Abbildung 8: Oberfläche

Von der Konfigurationsengine herzuleitende Größen werden durch das Schlüsselwort `CALC` gekennzeichnet und zusätzlich als `OUTPUT` oder `HIDDEN` spezifiziert. Berechnungen können durch Formeln, in Form von Entscheidungstabellen oder durch Abfragen formuliert werden.

In Abbildung 9 ist ein Beispiel für eine Entscheidungstabelle dargestellt. In der linken Spalte sind die Ergebnisse dargestellt, in allen anderen Spalten Bedingungen. Eine Entscheidungstabelle stellt eine Menge gleich strukturierter Regeln dar, jede Zeile steht für eine Regel. Die Bedingungen der Regeln werden von links nach rechts und von oben nach unten ausgewertet. Die erste Regel, deren Bedingungen alle erfüllt sind, „feuert“. Der unter `RESULT` aufgeführte Wert wird zurückgeliefert. Die Zahl der Spalten ist prinzipiell nicht begrenzt, es kann bei zu vielen Spalten jedoch unübersichtlich werden.

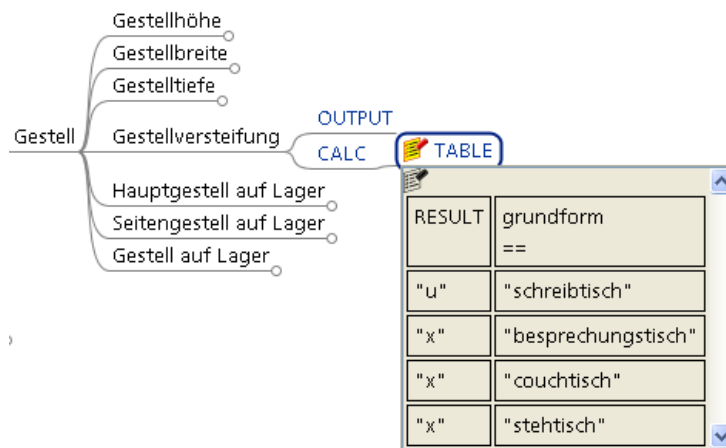


Abbildung 9: TABLE - Berechnung durch Entscheidungstabellen

Ein Beispiel für eine Abfrage finden Sie im Abschnitt 4.2 in Abbildung 11.

In K-Modell wird mit statischen und dynamischen Vorgabewerten gearbeitet. Dynamische Vorgabewerte werden analog zu den Berechnungen in Form von Ausdrücken oder Entscheidungstabellen formuliert. Gekennzeichnet werden sie durch das Schlüsselwort **DEFAULT**. Dem Benutzer wird der Wert vorgeschlagen, er kann ihn überschreiben. Vorgabewerte können dynamisch in Abhängigkeit von anderen Konfigurationsgrößen formuliert werden. Ändern sich die Vorbedingungen, so ändern sich auch die Vorgabewerte für Konfigurationsgrößen, die nicht vom Benutzer „berührt“ worden sind. Hat der Benutzer bereits den Vorgabewert überschrieben, so bleibt er erhalten.

K-Modell geht davon aus, dass die Fragen in beliebiger Reihenfolge vom Benutzer beantwortet werden können, soweit dies inhaltlich Sinn macht. Inhaltliche Abhängigkeiten zwischen Fragen lassen sich über Existenzbedingungen formulieren, die sich an der Oberfläche als „Sichtbarkeitsregeln“ auswirken. In Abbildung 10 ist ein Beispiel dargestellt.

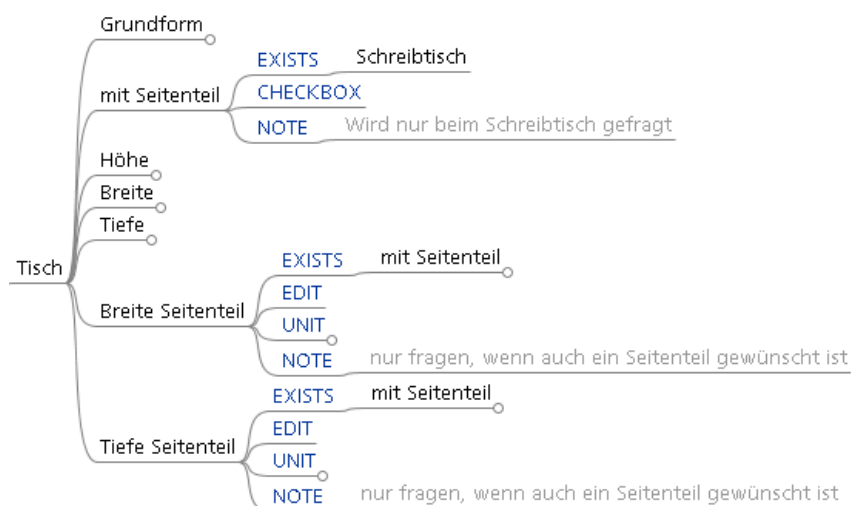


Abbildung 10: EXISTS - Existenzbedingungen

Das Beispiel in Abbildung 10 zeigt die Spezifikation des Seitenteils. Seitenteile gibt es nur als Option für Schreibtische. Nur wenn der Nutzer den Tischtyp „Schreibtisch“ wählt, bekommt er die Checkbox „mit Seitenteil“ angezeigt. Setzt er den Haken für das Seitenteil, so werden Breite und Tiefe des Seitenteils von ihm abgefragt.

4.2 Das Ergebnis

Das Ergebnis des Konfigurationsprozesses ist eine Vertriebsstückliste, die im K-Modell-Mindmap mit BOM gekennzeichnet ist. Die Stückliste kann beliebig tief strukturiert sein.

Wurde bei den Fragen zur hierarchischen Konfiguration bei einem Thema auf ein anderes Konfigurationsmodell verwiesen, so kann hier auf die daraus resultierende Vertriebsstückliste verwiesen werden.

Die Positionen der untersten Ebene werden entweder durch Auswahl aus einer Menge von Objekten oder durch Parametrierung eines Objekts aus der unter **CLASSES** spezifizierten Grundmenge besetzt.

Durch die Auswahl kann man einen PTO („pick to order“) oder einen ATO („assemble to order“) Prozess modellieren, bei dem auf vorgefertigte Produkte und/oder Baugruppen zurückgegriffen wird. Für MTO-Prozesse („make to order“), bei denen ein Produkt kundenindividuell gefertigt wird, passt die Modellierung mittels Parametrierung besser. Durch die Modellierung mittels Parametrierung kann man auch den Aufruf eines fertigungsorientierten Konfigurators darstellen.

Bei der Auswahl wird die Zuordnung durch eine Anfrage an eine Objektklasse angeben. In Abbildung 11 ist ein Beispiel gezeigt. Durch die Anfrage wird aus der Grundmenge der Tischgestelle das Gestell mit den angegebenen Eigenschaften ausgewählt.

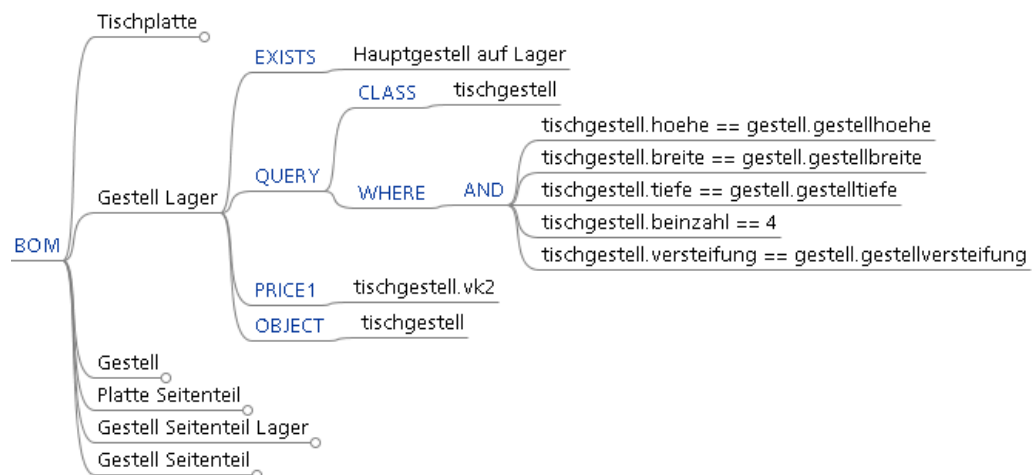


Abbildung 11: BOM - Definition der Stückliste

Die Klasse der Motoren muss mit all ihren Attributen unter **CLASSES** deklariert worden sein. Die Instanzen sind davon getrennt in den **INSTANCES** definiert. Durch die Zuordnung einer Instanz zu einer Stücklistenposition werden alle definierten Attribute durch die Attributswerte der ausgewählten Instanz mit Werten belegt.

Durch das Schlüsselwort **QTY** wird die Menge der Artikel spezifiziert. Ohne diese Angabe wird die Menge 1 angenommen. Zur Spezifikation können beliebige Ausdrücke analog zu **CALC** verwendet werden.

Die Stücklistenposition wird nur bei einer eindeutigen Zuordnung durch die Instanz der Grundmenge (= Artikel) ersetzt. Erfüllt entweder keiner oder mehr als ein Artikel die unter **WHERE** aufgeführten Anforderungen, so bleibt die Position unspezifiziert.

Besteht die Möglichkeit, dass die Anfrage auf mehrere Instanzen zutrifft, beispielsweise weil in der Anfrage mit **>** oder **<** verglichen wird, so kann zusätzlich eine Sortierung der Treffer mit **ORDER-BY** in Kombination mit **ASC** und **DESC** angeben. Durch das Schlüsselwort **FIRST** wird das erste Element der Sortierung ausgewählt.

Die Abbildung 12 gibt ein Beispiel für die Bestimmung einer Stücklistenposition durch Parametrierung. Die Argumente werden an ein Objekt der Klasse Tischgestell übergeben und dadurch dessen Merkmale gesetzt. In der Definition der Objektklasse unter `CLASSES` ist definiert, wie sekundäre Merkmale hergeleitet werden. Im Beispiel soll ausgedrückt werden, dass das Gestell mit den angegebenen Maßen kundenindividuell gefertigt wird.

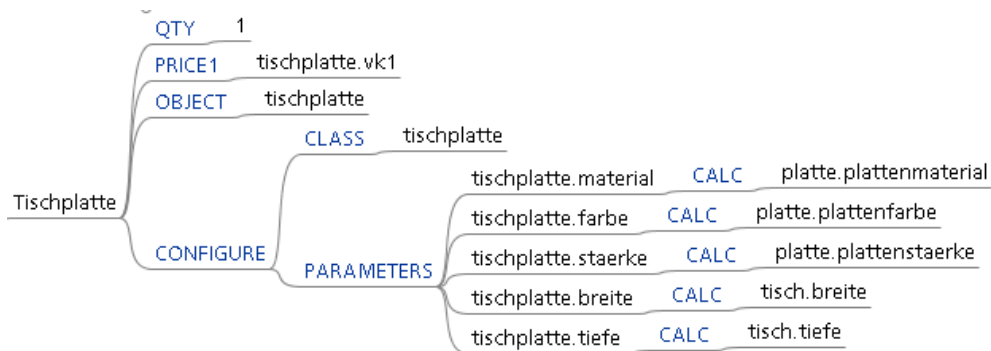


Abbildung 12: Zuordnung einer Stücklistenposition mittels Parametrierung

Soll eine Stücklistenposition nur in bestimmten Situationen erscheinen, so kann sie mit einer Existenzbedingung versehen werden. Nur wenn die Bedingung erfüllt ist, erscheint die Position in der Stückliste. Dies kann auf allen Stücklistenebenen genutzt werden.

Zwei Beispiele dafür finden Sie in Abbildung 11 und Abbildung 12. Die Positionen „Gestell Lager“ und „Gestell“ schließen sich aufgrund der Existenzbedingungen gegenseitig aus.

5. Der Katalog

Der Katalog ist der Startpunkt des Benutzers. Dort sucht der Benutzer nach geeigneten konfigurierbaren und voll spezifizierten Produkten. Der Katalog ist nach Kategorien strukturiert. Jede Kategorie kann weitere Unterkategorien oder Produkte enthalten. Ein Produkt kann in mehreren Kategorien enthalten sein, d.h. die Zuordnung Produkt – Kategorie ist nicht eindeutig. Der Benutzer kann ein Produkt auf mehreren Wegen finden. Abbildung 13 gibt ein Beispiel.

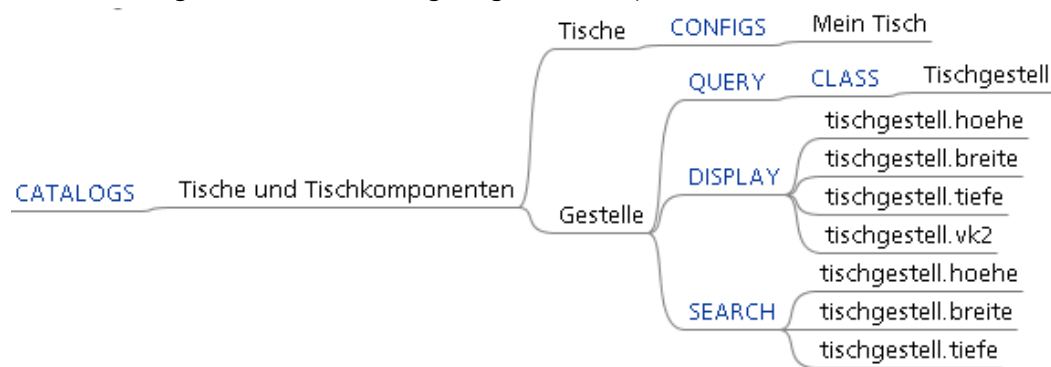


Abbildung 13: CATALOGS - Definition der Produktkataloge

In diesem Beispiel gibt es genau einen Katalog „Tische und Tischkomponenten“. Dieser Katalog hat die Kategorien „Tische“ und „Gestelle“. Die Kategorie „Tisch“ enthält nur ein konfigurierbares Produkt „Mein Tisch“. Die Kategorie „Gestelle“ enthält Instanzen der Klasse „Tischgestell“. Der Inhalt ist über die Anfrage spezifiziert, da diese keine WHERE Klausel enthält, werden alle Instanzen der Klasse angezeigt.

Zu jedem dargestellten Produkt / Artikel werden die mittels DISPLAY spezifizierten Attribute im Katalog dargestellt und können vom Benutzer als Auswahlkriterium genutzt werden. Über die mittels SEARCH spezifizierten Attribute kann der Benutzer gezielt nach bestimmten Eigenschaften suchen oder filtern. Es gelten hier Vererbungsmechanismen, die auf einer höheren Ebene für DISPLAY und SEARCH spezifizierten Attribute sind auch für die aktuelle Ebene verfügbar.

6. Der Modellierungsprozess

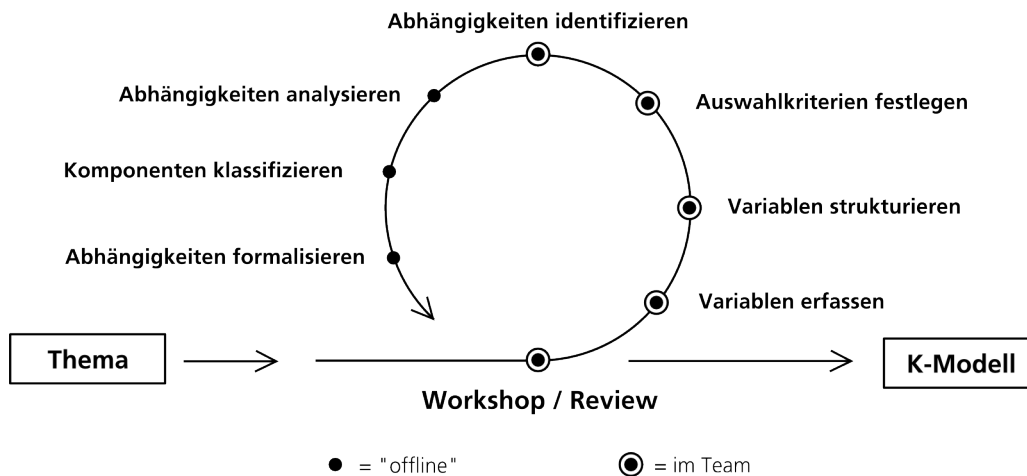


Abbildung 14: Der Modellierungszyklus

Der Modellierungsprozess zur systematischen Entwicklung von Konfigurationsmodellen besteht aus einem zyklischen Wechsel von Workshops im Team und Einzelarbeit. Ein Arbeitsthema wird in einem gemeinsamen Workshop initial erarbeitet und dann in Einzelarbeit „offline“ weiter ausgearbeitet. Diese Ausarbeitung wird wiederum in einem Workshop gemeinsam begutachtet und bei Bedarf weitergeführt. Dieser Zyklus wird wiederholt, bis das Thema vollständig ausgearbeitet ist. Die einzelnen Arbeitsschritte sind in Abbildung 14 dargestellt und werden im Folgenden erläutert.

6.1 Variablen erfassen

Konfigurationsgrößen werden am besten im Team in Workshops mit Brainstorming-Charakter erfasst. Bei derartigen Workshops geht es nicht darum, das Konfigurationsmodell im Detail festzulegen, sondern die Thematik in der Breite zu erfassen. Daher eignet sich eine informelle, mitlaufende Protokollierung in Form von Mindmaps sehr gut.

Folgende Sachverhalte werden für jede Konfigurationsgröße festgehalten.

- handelt es sich um eine Eingabegröße?
 - wird der Wert frei eingegeben?
 - wird der Wert aus einer Liste ausgewählt?
- handelt es sich um eine Ausgabegröße?

Für die Konfigurationsgröße wird eine Variable mit möglichst aussagekräftigem Namen im Mindmap angelegt und gegebenenfalls mit einem Kommentar versehen.

6.2 Variablen strukturieren

Thematisch zusammengehörende Variablen werden gruppiert. Um die Handhabbarkeit einer Klasse auch bei einer großen Zahl von Konfigurationsgrößen zu gewährleisten, bietet K-Modell die Möglichkeit, die Konfigurationsgrößen mittels sogenannter „Themen“ zu strukturieren.

K-Modell geht davon aus, dass die Konfigurationsgrößen ohne zusätzliche Präsentationsschicht unmittelbar an der Oberfläche präsentiert werden. Die Strukturierung in Themen kann als eine Präsentation der Konfigurationsgrößen auf verschiedenen Registerblättern interpretiert werden, die Konfigurationsgrößen eines Themas werden dem Benutzer „gleichzeitig“ in der im K-Modell definierten Reihenfolge präsentiert. Wird das Thema gewechselt, so werden die Konfigurationsgrößen des neuen Themas dargestellt, alle anderen sind „ausgeblendet“. Aus diesem Grund ist es wichtig, sich gegenseitig stark beeinflussende Konfigurationsgrößen in einem gemeinsamen Thema zu platzieren.

6.3 Auswahlkriterien festlegen

Wie in Abschnitt 1 dargelegt, geht die K-Modell Methode davon aus, dass durch das Konfigurationswissen Objekte aus den verfügbaren Komponenten ausgewählt oder parametrisiert werden. Die Produktstruktur legt fest, welche Klassen von Komponenten es gibt. In diesem Arbeitsschritt wird festgelegt, nach welchen Kriterien die Komponenten ausgewählt werden. Dazu werden sie als Merkmale der entsprechenden Klassen mit aussagekräftigen Namen angelegt und kommentiert.

6.4 Abhängigkeiten identifizieren

Abhängigkeiten zwischen den Konfigurationsgrößen bestimmen den Schwierigkeitsgrad der Lösung eines Konfigurationsproblems. Die Größen können sich auf verschiedene Arten gegenseitig beeinflussen:

- Berechnungen
Der Wert einer Variablen ist abhängig von anderen Variablen, die Berechnung ist beispielsweise als Formel definiert.
- Selektionskriterien
Komponenten werden aufgrund von Werten der Variablen ausgewählt.
- Vorgabewerte
Vorgabewerte können dynamisch in Abhängigkeit von anderen Variablen formuliert werden.
- Existenzbedingungen
Variablen existieren nur unter bestimmten Bedingungen, beispielsweise existiert in Abbildung 10 Breite und Tiefe des Seitenteils nur bei gesetzter Checkbox.

Diese Abhängigkeiten werden zuerst informell, ohne formale Spezifikation, erfasst.

6.5 Abhängigkeiten analysieren

Generell gilt, dass stark voneinander abhängige Konfigurationsgrößen auch thematisch zusammengehören. Andererseits soll die Zahl der Eingabefelder einer Maske eine bestimmte Grenze nicht überschreiten. Die Analyse der Abhängigkeiten gibt Hinweise darauf, welche Güte die gewählte Strukturierung der Konfigurationsgrößen hat.

6.6 Komponenten klassifizieren

Die zur Verfügung stehenden Komponenten werden erfasst und den Objektklassen zugeordnet. Bei ERP-Systemen spricht man bei dieser Tätigkeit von Klassifizierung der Stammdaten mit Merkmalsbewertung.

Im Konfigurationsprozess sollen merkmalsbasiert die geeigneten Komponenten gewählt werden können. Die Attribute repräsentieren die für diesen Zweck charakteristischen Eigenschaften, sie sollen die Komponenten hinsichtlich ihrer Eignung unterscheiden. Sollen die Komponenten automatisiert ausgewählt werden, dann muss es möglich sein, immer genau eine Komponente zu finden, mehrfache Treffer dürfen nicht auftreten. Dies bedeutet, dass es keine zwei Komponenten mit identischen Merkmalswerten geben darf. Ist dies doch der Fall, so bedeutet dies, dass entweder eine der beiden Komponenten überflüssig ist, da sie die gleichen Eigenschaften haben, oder dass ein Merkmal zur Unterscheidung fehlt. Im letzteren Fall ist weiteres Merkmal für diese Objektklasse anzulegen.

Komponenten mit identischen Merkmalswerten sind jedoch erlaubt, wenn die Komponenten interaktiv durch den Benutzer ausgewählt werden sollen. Dann muss die Information über die Unterschiede der Komponenten in beschreibenden Text oder in grafische Darstellungen untergebracht werden, die dem Benutzer angezeigt werden.

6.7 Abhängigkeiten formalisieren

Schließlich werden die Abhängigkeiten formal in Form von Formeln, Algorithmen und Auswahl-Statements aufgeschrieben. Dieser Schritt ist notwendig, um die Umsetzbarkeit in ein Software-Werkzeug zu gewährleisten.

Wichtig ist dabei, die informelle Beschreibung der Abhängigkeit parallel beizubehalten, um einerseits eine Kontrollmöglichkeit zu haben und andererseits die Kommunikation zu erleichtern.

7. Zusammenfassung

K-Modell ist eine softwareunabhängige Methode zur Entwicklung und zur Beschreibung von Konfigurationsmodellen in Form von speziell strukturierten Mindmaps. Sie basiert auf dem Ansatz, Strukturwissen, Wissen über Abhängigkeiten und Grunddaten getrennt zu beschreiben.

Die Methode zielt auf die Kommunikation des Konfigurationswissens zwischen den verschiedenen Fachabteilungen, von F&E über Produktmanagement, Vertrieb und Fertigung bis zur Geschäftsleitung, um die Inhalte diskutieren und festlegen zu können.

Typischerweise ist das Konfigurationswissen im Unternehmen auf eine Vielzahl von Köpfen verteilt. Mit K-Modell wird eine strukturierte Modellierungsmethode eingesetzt, bei der in Workshops das Konfigurationsmodell gemeinsam systematisch erarbeitet wird. Diese Vorgehensweise hat sich in einer Vielzahl von industriellen Projekten als effiziente Herangehensweise an Konfigurationsprobleme erwiesen.

8. Literatur

[Brinkop 1999] Axel Brinkop: "Variantenkonstruktion durch Auswertung der Abhängigkeiten zwischen den Konstruktionsbauteilen", Dissertationen zur Künstlichen Intelligenz, Band 204, Infix, St.-Augustin, 1999

[Brinkop 2009] Axel Brinkop: „K-Modell – Strukturierter Entwurf von Konfigurationsmodellen“, http://brinkop-consulting.com/paper/k_modell_2009.pdf

[Freeplane 2014] <http://freeplane.sourceforge.net>