

# Fachkonzept eines konfigurativen Referenzmodellierungswerkzeugs

Patrick Delfmann, Armin Stein

**Abstract:** *Die konfigurative Referenzmodellierung stellt ein probates Mittel zur Verwaltung von Varianten konzeptioneller Softwaremodelle dar. Eine adäquate Unterstützung der Softwareentwicklung durch die konfigurative Referenzmodellierung kann jedoch nur dann gewährleistet werden, wenn ein entsprechendes Modellierungswerkzeug zur Verfügung steht. Die konzeptionellen Grundlagen der konfigurativen Referenzmodellierung werden in diesem Beitrag eingeführt und auf das konkrete Problem einer Werkzeugimplementierung übertragen. Es wird ein Fachkonzept in Form eines Datenmodells vorgestellt, das den Ausgangspunkt für die Implementierung eines konfigurativen Referenzmodellierungswerkzeugs darstellt. Instanzen der einzelnen Elemente des Datenmodells repräsentieren hierbei den Inhalt der durch das Modellierungswerkzeug zu verwaltenden Modelle und Modellvarianten.*

## 1 Wiederverwendungsansätze für Informationsmodelle

Die Effektivität und Effizienz der fachkonzeptionellen Modellierung kann erhöht werden, indem bestehende Modelle als Ausgangslösung für die Entwicklung spezifischer Modelle – z. B. kundenspezifischer Softwaremodelle – genutzt werden [Schü98, S. 367ff.]. In einer entsprechenden Modellbeziehung werden als Ausgangslösung dienende Modelle als *Referenzmodelle* bezeichnet. Für die Ableitung spezifischer Modelle eignen sich unterschiedliche Ansätze zur Wiederverwendung fachkonzeptioneller Modelle [Broc03, S. 259ff.], die auch kombiniert eingesetzt werden können. In vielen Fällen ist eine Anpassung des Referenzmodells durch *freie Modifikation* seiner Modellteile vorgesehen [Krus96; Sche97; Kurb03; Hamm97; Hars94; Rohl95; BeSc04]. Der Detaillierungsgrad des Referenzmodells wird in diesem Fall bewusst eingeschränkt. Ist das Referenzmodell in Komponenten zerlegt, können diese Komponenten im Rahmen der

*Aggregation* [Lang97; Remm97; Hamm99; HaPS99; Schu01] zu neuen Lösungen zusammengesetzt werden. Die Kombinierbarkeit kann dabei durch Schnittstellendefinitionen eingeschränkt werden. Die *Instanziierung* eines Referenzmodells [Schü98; Schw99; Schu01; Wolf01] sieht dagegen vor, dass Platzhalter durch zulässige Ausprägungen ihres Wertebereichs gefüllt werden, wodurch die Anpassung des Modells explizit beschränkt wird.

*Konfigurierbare* Referenzmodelle enthalten Regeln, die determinieren, wie die Modelle in Abhängigkeit anwendungskontextspezifisch gewählter Ausprägungen von Parametern – so genannten *Konfigurationsparametern* – zu verändern sind [BDKK02; BDDK03; BeDK04; Delf06; Knac06; DeKn07]. Bei der Konfiguration werden entsprechend den Konfigurationsregeln aus einem Gesamtmodell, das aus mehreren spezifischen Modellvarianten besteht, einzelne spezifische Modelle abgeleitet. Der Konfigurationsprozess entspricht damit einer Projektion des Ursprungsmodells, die eine reduzierte Version des Ursprungsmodells zum Ergebnis hat. Anwendungskontextspezifisch (z. B. kundenspezifisch) nicht relevante Modellbestandteile werden ausgeblendet.

Die Konfiguration eignet sich in besonderem Maße für die Formulierung von konkreten Gestaltungsempfehlungen und für die Anpassung des Referenzmodells an vorausgeplante Anwendungskontexte. Damit ist sie insbesondere für die Veraltung von Modellvarianten im Rahmen der Softwareentwicklung geeignet, da die Modellvarianten Softwareanforderungen von Kunden repräsentieren, welche im Vorfeld bekannt sein sollten (vgl. auch den einführenden Beitrag von JÖRG BECKER, PATRICK DELFMANN und TOBIAS RIEKE in diesem Band).

Die übrigen Wiederverwendungsansätze zeichnen sich gegenüber der Konfiguration dadurch aus, dass die Vorwegnahme der im Rahmen der Referenzmodell-anwendung notwendig werdenden Modellanpassungen geringer ausfällt. Anwendern der übrigen Wiederverwendungsansätze werden damit größere kreative Gestaltungsfreiräume bei der Anpassung von Modellen eingeräumt, weswegen sich diese Ansätze insbesondere für Feinanpassungen eignen, die sich bspw. an eine Konfiguration anschließen.

## **2 Grundlagen der konfigurativen Referenzmodellierung**

### **2.1 Konfigurationsparameter**

Die Ursachen für die Konstruktion von Softwaremodellvarianten, die mithilfe der konfigurativen Referenzmodellierung verwaltet werden sollen,

bilden abweichende Anforderungen unterschiedlicher Nutzergruppen. Diese Anforderungen können einerseits aus unterschiedlichen betriebswirtschaftlichen Merkmalen der Unternehmen, die die Software nutzen, resultieren. Andererseits können persönliche Präferenzen hinsichtlich der Gestaltung der Software dazu führen, dass unterschiedliche Anforderungen an eine Software formuliert werden.

Die unterschiedlichen Anforderungstypen dienen der Konfiguration als Konfigurationsparameter und werden in *Unternehmensmerkmale* und *Perspektiven* gegliedert [BDKK02, S. 38]:

- *Unternehmensmerkmale* und ihre *Ausprägungen* beschreiben die Klasse von Unternehmen, für die eine Anpassung des Referenzmodells durchgeführt werden soll. Beispiele für Unternehmensmerkmale in der Handelsdomäne sind *Geschäftsart* oder *Wirtschaftsstufe* [BeSc04, S. 2]. Exemplarische Ausprägungen der Geschäftsart sind *Lager-*, *Strecken-* und *Zentralregulierungsgeschäft*. In der Industriedomäne werden derartige Unterscheidungen bspw. durch Betriebstypologien [MeLo00, S. 116ff.; LoHM02, S. 279ff.; Kurb03, S. 32ff.] vorgenommen.
- *Perspektiven* werden vorrangig durch den verfolgten *Modellierungszweck*, z. B. *Anwendungssystem-* oder *Organisationsgestaltung*, die *Rolle* der Anwender im Projekt, z. B. *Methodenexperte* oder *Anwender*, und *sonstige Einflüsse* wie z. B. *layouttechnische Präferenzen* von Modellanwendern determiniert [BDKK02, S. 28ff.; RoSD05, S. 50ff.].

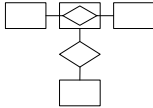
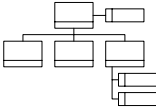
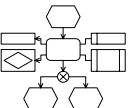
## 2.2 Konfigurationsmechanismen

Die Umsetzung der Konfiguration, d. h. die Anpassung der Modelle gemäß den Anforderungen, die in den Konfigurationsparametern repräsentiert sind, wird – dem Prinzip der Effizienzsteigerung folgend – durch automatisierte Konfigurationsmechanismen realisiert. Um die Anwendung der Konfigurationsmechanismen benutzerfreundlich und einfach zu gestalten, ist es vorteilhaft, Mechanismen mit unterschiedlich weitem Wirkungsgrad bereitzustellen. Es werden deshalb fünf unterschiedliche Kategorien von Konfigurationsmechanismen differenziert (vgl. im Folgenden [BDKK02, S. 72ff.]):

### **Modelltypselektion**

Werden Modelle mit der gleichen Modellierungssprache konstruiert, so gehören sie dem gleichen *Modelltyp* an, z. B. dem der *Ereignisgesteuerten Prozessketten* (EPK, vgl. [KeNS92]), der *Fachbegriffsmodelle* [Spie93],

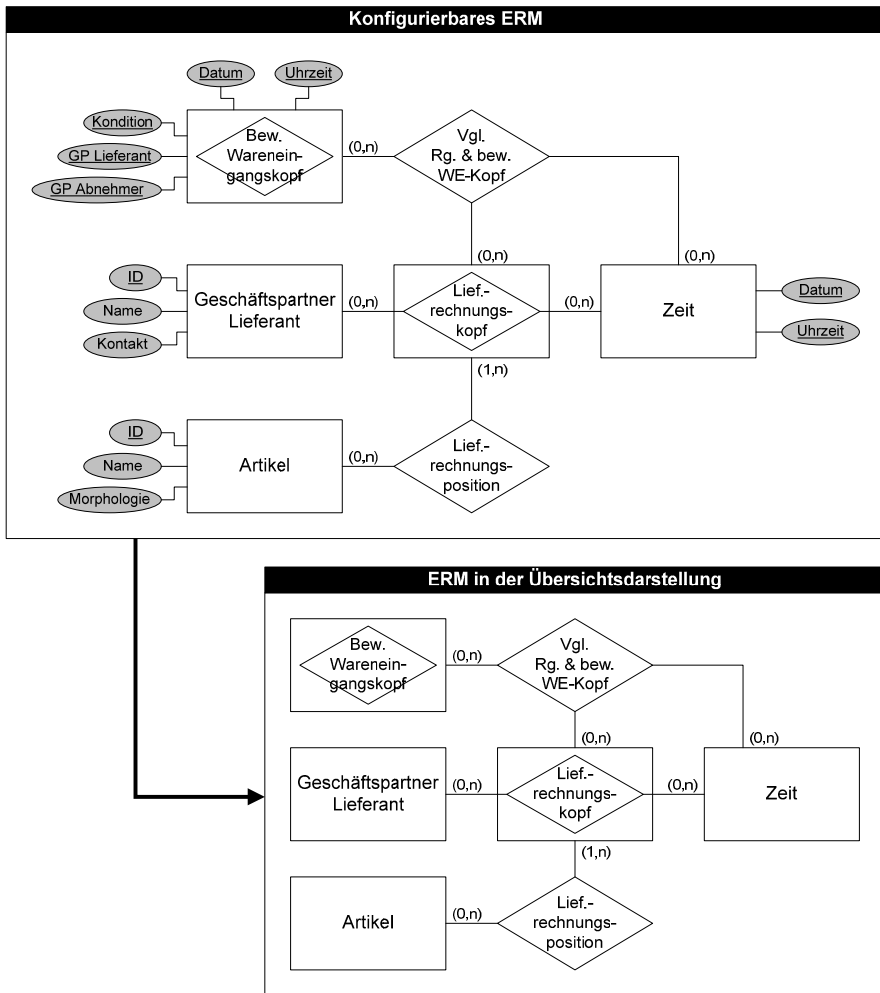
der *Entity-Relationship-Modelle* (ERM, vgl. [Chen76]), oder der *Organigramme* (z. B. [Groc82, S. 305; Schr96, S. 125]). Der Konfigurationsmechanismus der Modelltypselektion unterstützt die Zuordnung von Modelltypen zu Konfigurationsparametern, wodurch eine Nutzergruppenabhängige Ein- und Ausblendung von Modellen bestimmter Typen ermöglicht wird. In der Softwareentwicklung ist dieses Vorgehen sinnvoll, wenn das Softwarefachkonzept aufgabenteilig erstellt wird und bspw. Datenbankentwicklern ausschließlich die Datensicht zur Verfügung stehen soll (vgl. Tabelle 1).

Modelltyp	Perspektive Geschäftslogik	Perspektive Datenbank
Entity-Relationship-Modell		
Organigramm		
Ereignisgesteuerte Prozesskette		

**Tabelle 1:** Exemplarische Modelltypselektion

### **Elementtypselektion**

Elementtypselektionen ermöglichen es, zu Modelltypen Varianten zu bilden, die sich in der Menge der verwendbaren Modellelementtypen unterscheiden. Je nach eingenommener Perspektive des Modellanwenders kann es erforderlich sein, Modelltypen mit unterschiedlicher Ausdrucksmächtigkeit bzgl. des Angebots an Elementtypen zur Verfügung zu stellen. Beispielsweise unterscheiden sich Varianten der erweiterten Ereignisgesteuerten Prozesskette in den Modellelementtypen, die an Funktionen annotiert werden können. Kandidaten für die Annotation stellen z. B. Entitytypen und Organisationseinheiten dar. Die Modelltypvarianten werden gebildet, indem ihnen Elementtypen abhängig vom Konfigurationsparameter zugeordnet werden. Im Rahmen der Softwareentwicklung unterstützen Elementtypselektionen den Entwicklungsprozess bspw. dann, wenn aus Übersichtsgründen Detailbereiche von Modellen ausgeblendet werden müssen (etwa Attributbeschreibungen in ERMs; vgl. Abbildung 1).



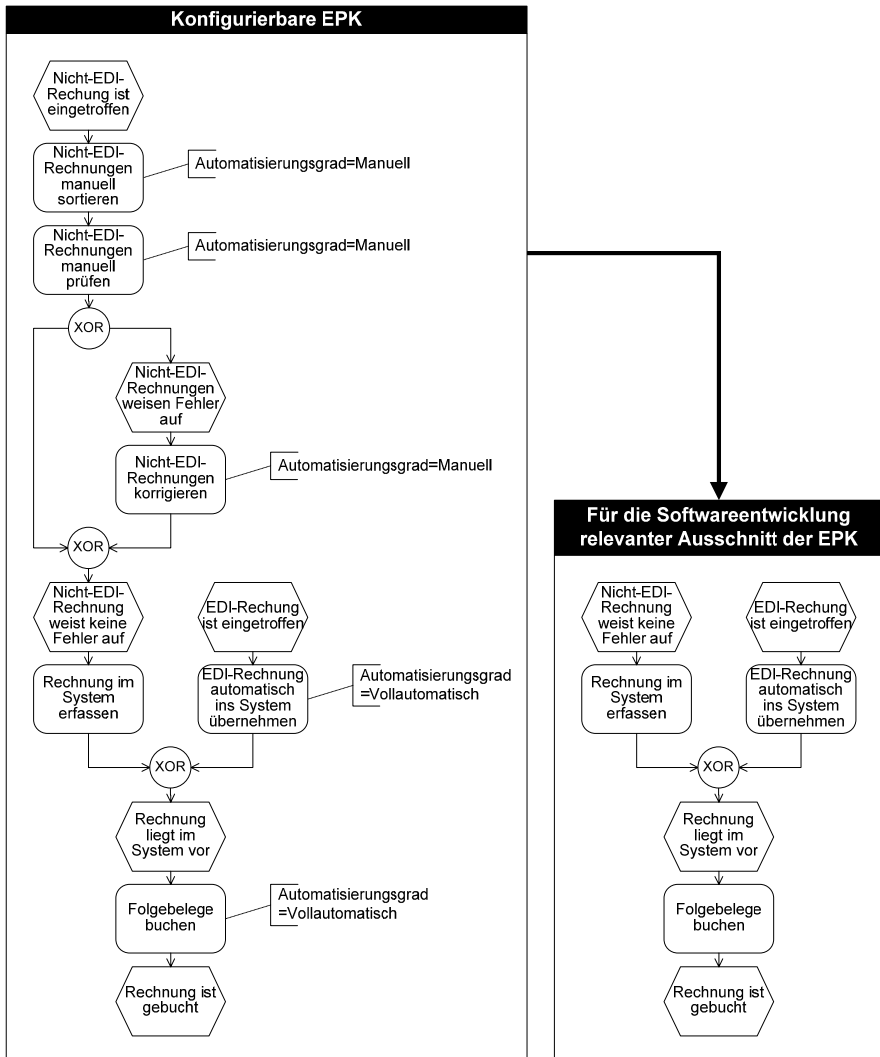
**Abbildung 1:** Exemplarische Elementtypselektion im ERM [Delf06, S. 102]

### Elementselektion

Elementselektionen äußern sich in der konfigurationsparameterabhängigen Selektion von einzelnen Modellelementen, bspw. einer einzelnen Prozessmodellfunktion „Rechnung prüfen“. Auf Basis dieser Selektion können einzelne Elemente ausgeblendet werden, die für den Anwendungskontext des aktuellen Konfigurationsparameters nicht relevant sind. Im Rahmen der Elementselektion werden zwei Konfigurationsmechanismen unterschieden:

### Elementselektion über Konfigurationsattribute

Die Variantenbildung mithilfe der Elementselektion über Konfigurationsattribute erfolgt über die Auswertung von Eigenschaften, die den Modell-elementen zugeordnet sind. Beispielsweise kann einer Funktion eines Prozessmodells ein Attribut zugeordnet werden, das diese anhand ihres Automatisierungsgrads klassifiziert.

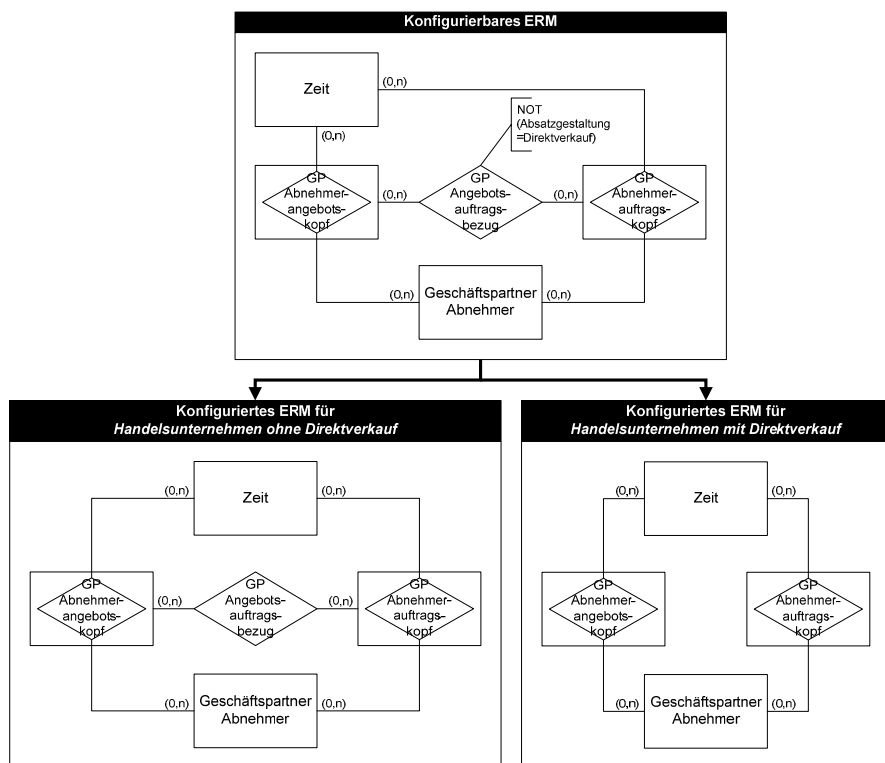


**Abbildung 2:** Exemplarische Elementselektion über Konfigurationsattribute in der EPK [BDKK02, S. 111]

Konfigurationsparameterspezifisch kann mithilfe des Mechanismus festgelegt werden, welche Funktionen mit welchem Automatisierungsgrad angezeigt bzw. ausgeblendet werden sollen. Eine solche Klassifikation wird bspw. vorgenommen, wenn die zur Verfügung stehenden Modelle im Rahmen einer integrierten Anwendungssystem- und Organisationsgestaltung entstanden sind und anhand des Automatisierungsgrads ermittelt werden kann, ob die Modellteile sich für eine Implementierung – in diesem Fall im Rahmen der Softwareentwicklung – eignen oder nicht (vgl. Beispiel in Abbildung 2).

### Elementselektion über Konfigurationsterme

Die Elementselektion über Konfigurationsterme ist dadurch gekennzeichnet, dass dem zu konfigurierenden Modellelement ein boolescher Ausdruck zugewiesen wird, der die direkte Zuordnung des Elements zu einer oder mehreren Konfigurationsparameterausprägungen ermöglicht.



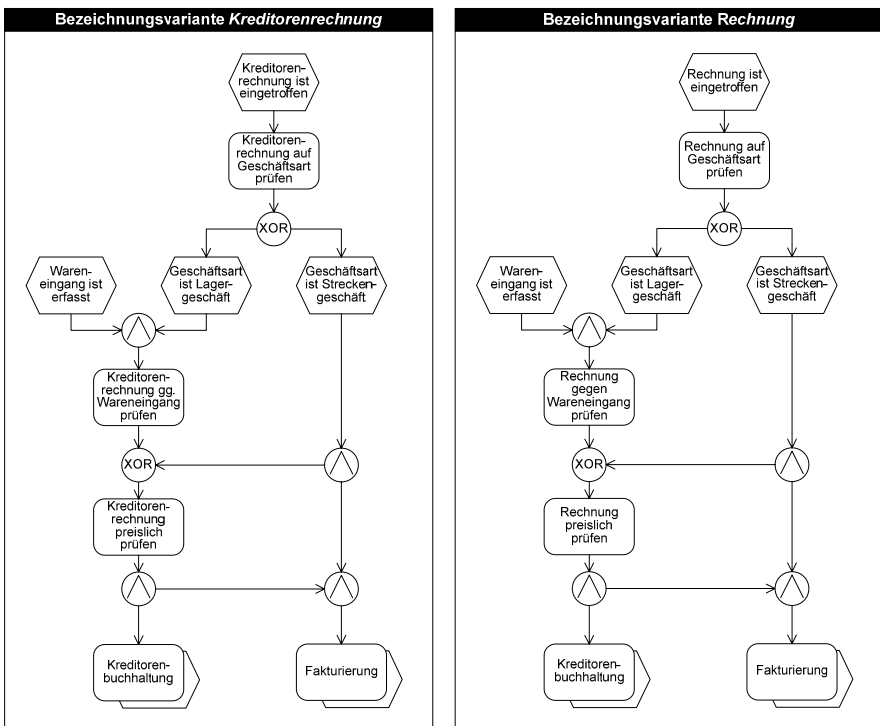
**Abbildung 3:** Exemplarische Elementselektion über Konfigurationsterme im ERM [BDKK02, S. 117; Delf06, S. 121]

Diese Vorgehensweise bietet sich im Software Engineering insbesondere für die Unterscheidung von Kundenvarianten an (vgl. exemplarisch Abbildung 3; hier wird ein Datenmodell angepasst, je nachdem, ob das softwarenutzende Handelsunternehmen Direktverkauf betreibt oder nicht, was sich in der Notwendigkeit widerspiegelt, einen Bezug zwischen Angebot und Auftrag herzustellen zu müssen oder nicht).

Die Wirkungsweise der Selektion unterscheidet sich nicht von derjenigen der Elementselektion über Konfigurationsattribute. Abhängig von der jeweiligen Konfigurationssituation kann die Anwendung eines der beiden Konfigurationsmechanismen gegenüber dem anderen zu Effizienzsteigerungen bei der Konstruktion und Anwendung der konfigurativen Modelle führen (vgl. hierzu Abschnitt 3.2).

### Bezeichnungsvariation

Der Mechanismus der Bezeichnungsvariation berücksichtigt, dass es erforderlich sein kann, die Bezeichnungen von Modellelementen auszutauschen.



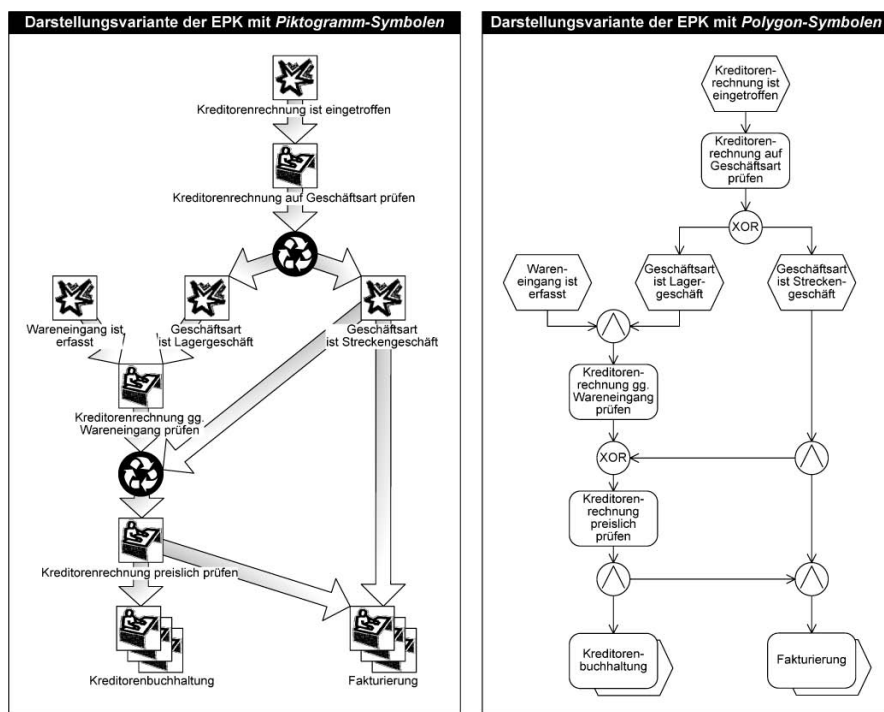
**Abbildung 4:** Exemplarische Bezeichnungsvariationen in der EPK [DeKn07]



Im Software Engineering kommt sie vorwiegend bei der Konstruktion von Kundenvarianten zum Einsatz, wenn sich in den Kundenunternehmen unterschiedliche Begriffskonventionen etabliert haben. So wird eine Kreditorenrechnung in unterschiedlichen Unternehmen bspw. synonym als „Kreditorenrechnung“ oder in einfacherer Form als „Rechnung“ bezeichnet (vgl. Abbildung 4).

### Darstellungsvariation

Die Darstellungsvariation dient dazu, abhängig von der jeweiligen Nutzergruppe der Modelle die grafischen Symbole der einzelnen Modellelemente auszutauschen [Allw98]. Im Software Engineering kann die Darstellungsvariation dazu eingesetzt werden, zum Zweck der Diskussion der Modelle mit dem Kunden diese anschaulicher zu gestalten, indem die häufig abstrakten Symbole in Informationsmodellen durch anschaulichere Bildzeichen ausgetauscht werden (vgl. Abbildung 5).



**Abbildung 5:** Exemplarische Darstellungsvarianten der EPK [DeKn07]

### 3 Fachkonzept des Referenzmodellierungswerkzeugs

Das Fachkonzept des konfigurativen Referenzmodellierungswerkzeugs basiert auf theoretischen Grundlagen der konfigurativen Referenzmodellierung, die in der Literatur ausführlich diskutiert worden sind [BDKK02; BKKD03; BeDK04; Delf06; DeKn07]. Im Rahmen dieser Beiträge werden Konzepte zur Konfiguration von Informationsmodellen eingeführt, die mit dem Ziel der Übertragbarkeit auf beliebige Modellierungssprachen und Implementierungsplattformen allgemeingültig formuliert werden. Darüber hinaus sind sie sprachtheoretisch ausgerichtet und werden mithilfe mehrerer sprachlicher Modellebenen spezifiziert. Diese Konzepte werden im Folgenden auf den konkreten Anwendungskontext der Implementierung als Modellierungswerkzeug übertragen.

Das werkzeugbezogene Fachkonzept lehnt sich an die *Architektur Integrierter Informationssysteme (ARIS)* [Sche97]) an, welche sich zur Beschreibung fachkonzeptioneller Inhalte durchgesetzt hat. Aus den Sichten der Architektur, *Daten, Prozesse, Funktionen* und *Organisation*, wird insbesondere die Datensicht fokussiert, da die Verwaltung von Modellinhalten sowie Konfigurationseinstellungen auf Datenebene stattfindet.

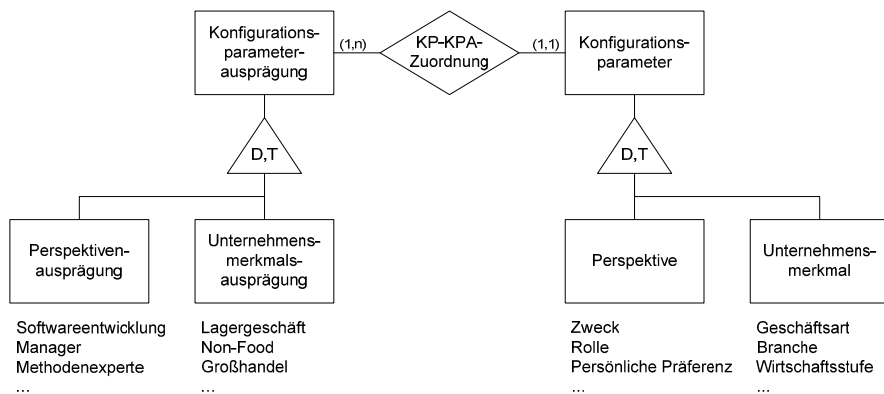
Auf die umfangreiche Beschreibung von Prozesslogik, die insbesondere für Zwecke der grafischen Darstellung von Modellen auf der Modellierungsoberfläche des Werkzeugs sowie für die allgemeine Bedienung notwendig ist, wird verzichtet (vgl. hierzu auch den Beitrag von TOBIAS RIEKE und ARMIN STEIN in diesem Band). Für die Manipulation der Daten zur Realisierung von Konfigurationsmechanismen reichen deklarative Mechanismen, wie sie bspw. von einfachen Datenbankmanipulationssprachen (z. B. SQL) zur Verfügung gestellt werden, weitestgehend aus.

Die fachkonzeptionellen Spezifikationen werden als erweiterte Entity-Relationship-Modelle in der Notation mit (min,max)-Kardinalitäten nach SCHLAGETER und STUCKY dargestellt [ScSt83, S. 90f.].

#### 3.1 Grundlegende Spezifikationen

Grundlage der konfigurativen Verwaltung von Informationsmodellen sind die in Abschnitt 2.1 eingeführten *Konfigurationsparameter* mit ihren jeweiligen *Ausprägungen*, die determinieren, ob Modellinhalte für entsprechende Nutzergruppen relevant sind oder nicht (vgl. Abbildung 1). Ein Beispiel für eine Kombination aus Konfigurationsparameter und Konfigurationsparameterausprägung ist die Zuordnung „Geschäftsart=Lagergeschäft“, wobei der Konfigurationsparameter durch die *Geschäftsart* und die Konfigurationsparameterausprägung durch das *Lagergeschäft* beschrie-

ben wird. Ein weiteres Beispiel für solch eine Kombination ist die Aussage „Rolle=Manager“. Im ersten Fall ist der Konfigurationsparameter als *Unternehmensmerkmal* und seine Konfigurationsparameterausprägung als *Unternehmensmerkmalsausprägung* spezialisiert. Im zweiten Fall wird eine *Perspektive* mit einer *Perspektivenausprägung* in Beziehung gesetzt. Weitere Beispiele für Unternehmensmerkmale, Unternehmensmerkmalsausprägungen, Perspektiven und Perspektivenausprägungen finden sich in Abbildung 6 sowie in Abschnitt 2.1.

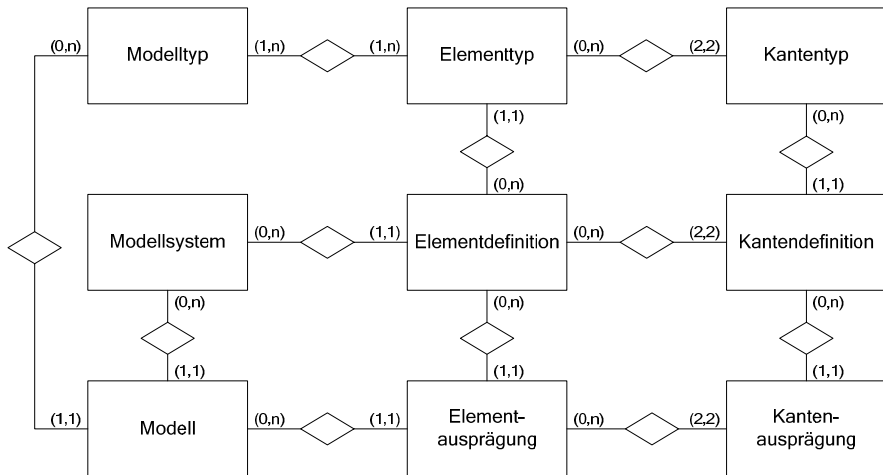


**Abbildung 6:** Konfigurationsparameter

Weitere grundlegende Spezifikationen des Modellierungswerkzeugs sind bezüglich der Verwaltung der allgemeinen Modelldaten vorzunehmen, d. h. es ist darzustellen, wie Modelle, Modellierungssprachen und Modellelemente zusammenhängen (vgl. Abbildung 7). Jedes *Modell*, das mit dem Modellierungswerkzeug verwaltet wird, gehört eindeutig zu einem *Modellsystem*. In Modellsystemen können mehrere Modelle enthalten sein, die auch in unterschiedlichen Modellierungssprachen verfasst sein (bspw. als EPK oder als ERM), also unterschiedlichen *Modelltypen* angehören können.

Innerhalb eines Modellsystems werden die enthaltenen Modelle – dem Grundsatz der Architektur Integrierter Informationssysteme folgend – miteinander verknüpft (bspw. wenn ein Prozess, der durch ein Prozessmodell repräsentiert wird, auf Daten, die durch Datenmodelle repräsentiert werden, zugreift). Die Verknüpfung wird über die Modellelemente realisiert (bspw. konkrete Entitytypen, Funktionen oder Ereignisse), deren *Elementdefinition* eindeutig im Modellsystem angelegt wird. Diese Elementdefinitionen lassen sich in verschiedenen Modellen als *Elementausprägungen* wiederverwenden. Wird bspw. in einem Datenmodell in ERM-Notation ein Entitytyp „Auftrag“ definiert, so werden eine Elementdefinition und

eine entsprechende Elementausprägung für das Datenmodell angelegt. Soll dieser „Auftrag“ in einem Prozessmodell wiederverwendet werden – etwa als Input für eine Prozessfunktion – so wird für das Prozessmodell eine weitere Elementausprägung angelegt, die auf die bereits bestehende Elementdefinition verweist. Auf diese Weise ist sichergestellt, dass das wiederverwendete Modellelement semantisch demjenigen im Ursprungsmodell entspricht. Es ist eine echte Verknüpfung hergestellt.



**Abbildung 7:** Grundlegendes Datenmodell des Modellierungswerkzeugs

Modellelemente bzw. Elementdefinitionen gehören eindeutig einem *Elementtyp* an, der wiederum einem oder mehreren Modelltypen zugeordnet sein kann. Beispiele für Elementtypen sind „Funktion“, „Ereignis“ oder „Entitytyp“, während Beispiele für Elementdefinitionen sowie Elementausprägungen „Funktion: Rechnung prüfen“ oder „Entitytyp: Lieferant“ sind.

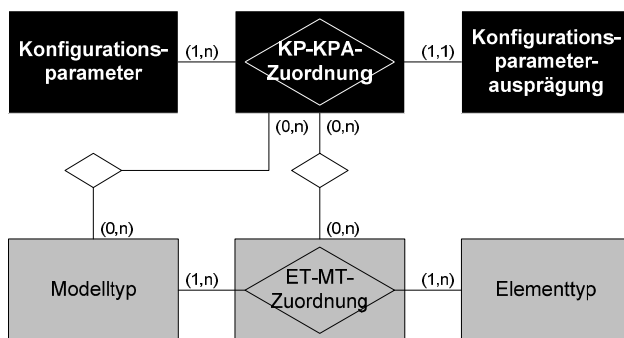
Modellelemente werden in Modellen mit Kanten in Beziehung gesetzt. Bspw. wird die zeitlich-sachlogische Abfolge von Funktionen innerhalb eines Prozessmodells durch gerichtete Kontrollflusskanten von Aktivität zu Aktivität repräsentiert. Analog zu Elementdefinitionen und -ausprägungen werden auch bei Kanten *Kantendefinitionen* und -ausprägungen unterschieden. Die Kardinalität (2,2) stellt sicher, dass eine Kante stets mit genau zwei Modellelementen verbunden sein muss. Ebenso gehören Kantendefinitionen jeweils genau einem *Kantentyp* an, der festlegt, zwischen Elementen welcher Elementtypen die Kante gezogen werden darf.

## 3.2 Konfigurationsmechanismen

Die grundlegenden Spezifikationen, die die Modelldatenbasis darstellen, werden zur Realisierung der Konfigurationsmechanismen mit den Spezifikationen der Konfigurationsparameter verknüpft. Um Modellinhalte abhängig von Konfigurationsparametern darstellen zu können, werden die Konfigurationsparameter und deren Ausprägungen im Datenmodell mit den Elementen der Modelldatenbasis in Beziehung gesetzt. Das Modellierungswerkzeug kann dann zur Darstellung der Modellinhalte auf eben diese Beziehungen zugreifen.

### *Modelltypselektion und Elementtypselektion*

Zur Realisierung der Modelltypselektion werden im Datenmodell Modelltypen mit der Kombination von Konfigurationsparametern und Konfigurationsparameterausprägungen (*KP-KPA-Zuordnung*) verknüpft (vgl. Abbildung 8; bereits eingeführte Elementtypen werden im Folgenden grau schattiert; die Konfigurationsparameterstruktur wird aufgrund ihrer zentralen Bedeutung für die Konfiguration schwarz hervorgehoben). Im entsprechenden Relationshiptyp werden zulässige Modelltypen für die jeweilige Konfigurationsparameterausprägung vermerkt. Ist ein Modelltyp nicht vermerkt, so werden alle Modelle, die diesem Modelltyp angehören, für die entsprechende Konfigurationsparameterausprägung ausgeblendet.



**Abbildung 8:** Modelltypselektion und Elementtypselektion

Elementtypselektionen werden analog realisiert. Um Elementtypselektionen pro Modelltyp realisieren zu können, wird die Zuordnung zur KP-KPA-Zuordnung nicht über den Elementtyp selbst, sondern über seine Beziehung zum jeweiligen Modelltyp vorgenommen (*ET-MT-Zuordnung*). So ist es möglich, Elementtypen aus einem Modelltyp zu entfernen, während sie für einen anderen Modelltyp weiterhin zur Verfügung stehen.

## **Modell- und Elementselektion – Überblick**

Wie bereits in Abschnitt 2.2 angedeutet werden Elementselektionen mit zwei unterschiedlichen Formen von Konfigurationsmechanismen umgesetzt. Zur Annotation von Konfigurationseinstellungen an Modellelemente werden spezielle Platzhalter eingeführt, die *Konfigurationsattribute* und *Konfigurationsterme* repräsentieren und als *Konfigurationsannotationen* diese beiden Konfigurationseinstellungsformen zusammenfassen. Um sowohl Modellelementen als auch ganzen Modellen Konfigurationseinstellungen annotieren zu können, wird die Konfigurationsannotation sowohl dem *Modell* als auch der *Elementausprägung* zugewiesen (vgl. hierzu und im Folgenden Abbildung 9).

## **Modell- und Elementselektion über Konfigurationsattribute**

Die Spezialisierung Konfigurationsattribut der Konfigurationsannotation setzt sich aus einem *Attribut* sowie seiner *Ausprägung* zusammen. Ein Beispiel für ein konkretes Attribut ist der „Automatisierungsgrad“, der bspw. eine Funktion in einem Prozessmodell charakterisiert. Eine korrespondierende exemplarische Attributausprägung ist „vollautomatisch“. Das entsprechende Konfigurationsattribut ist „Automatisierungsgrad=vollautomatisch“, das sich aus Attribut und Ausprägung zusammensetzt. Durch die Zuweisung solcher Konfigurationsattribute lassen sich Modellelemente in Kategorien einordnen, auf deren Grundlage eine Konfiguration vorgenommen werden kann.

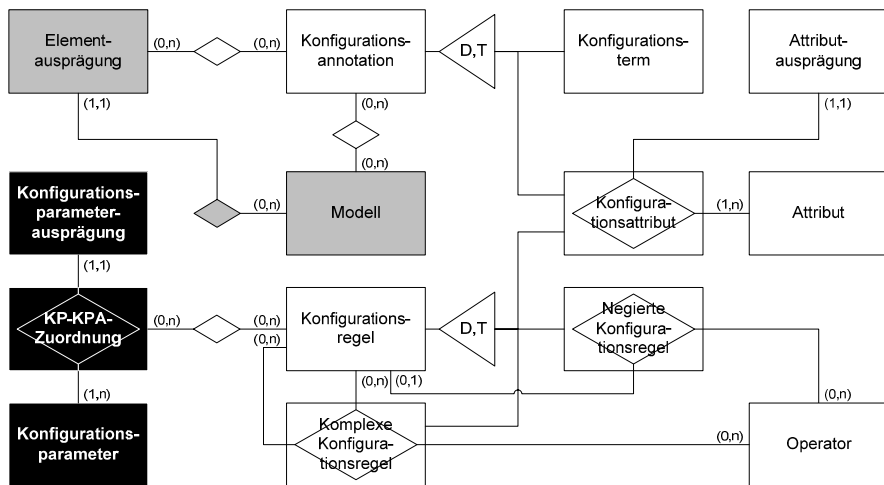
Die Konstruktion von Modell- und Modellelementkategorien lässt sich anhand von einzelnen Konfigurationsattributen und darüber hinaus anhand von logisch verknüpften Mengen von Konfigurationsattributen vornehmen. Deshalb werden sowohl einzelne als auch zusammengesetzte Konfigurationsattribute unter dem Begriff *Konfigurationsregeln* zusammengefasst. Eine Konfigurationsregel, die sich aus logisch verknüpften, mehreren Konfigurationsattributen zusammensetzt, wird als *komplexe Konfigurationsregel* spezialisiert. Eine komplexe Konfigurationsregel enthält stets genau wiederum zwei Konfigurationsregeln, die über einen logischen *Operator* (*AND* bzw. *OR*) verknüpft sind. Sollen mehr als zwei Konfigurationsregeln verknüpft werden, so lassen sich über die Generalisierung der komplexen Konfigurationsregel zur *Konfigurationsregel* verschachtelte logische Ausdrücke von Konfigurationsattributen aufbauen. Zusätzlich kann einer Konfigurationsregel ein Operator direkt zugewiesen werden – nämlich genau dann, wenn die Konfigurationsregel durch einen *NOT*-Operator negiert werden soll (vgl. *negierte Konfigurationsregel*). Negierte Konfigurationsregeln können analog zu komplexen Konfigurationsregeln über die Gene-

realisierung zur *Konfigurationsregel* verschachtelt werden. Ein Beispiel für eine negierte Konfigurationsregel, die eine komplexe Konfigurationsregel enthält, ist die folgende:

*NOT (Automatisierungsgrad=vollautomatisch AND  
Detailelement=TRUE)*

Sie drückt aus, dass nur die Modelle bzw. Modellelemente angezeigt werden sollen, die nicht der Kombination aus den Eigenschaften „vollautomatisch“ und „Detailelement“ entsprechen – d. h., die entweder nicht vollautomatisiert sind oder nicht die Eigenschaft eines Detailelements besitzen oder weder vollautomatisiert sind noch Detailinformationen vermitteln.

Ob eine Konfigurationsregel angewendet wird, ist abhängig von der jeweiligen Nutzergruppe, die auf die Modelle zugreift. Deshalb werden Konfigurationsregeln mit der KP-KPA-Zuordnung in Beziehung gesetzt.



**Abbildung 9:** Element- und Modellselektion über Konfigurationsattribute

Ein Konfigurationsprozess, der durch die Elementselektion über Konfigurationsattribute umgesetzt wird, läuft damit wie folgt ab:

- Anhand der ausgewählten Konfigurationsparameterausprägung wird ermittelt, welche Konfigurationsregeln auszuführen sind.
- Die Konfigurationsregeln werden mit den Konfigurationsattributen verglichen, die den Modellen und Modellelementen annotiert sind.
- Entsprechen die Konfigurationsattribute an den Modellen oder Modellelementen den Konfigurationsregeln, so werden die Modelle oder Modellelemente angezeigt. Anderenfalls werden sie ausgeblendet. Im Bei-

spielfall würde etwa ein Modellelement, das das Konfigurationsattribut „Automatisierungsgrad=manuell“ trägt, angezeigt.

### Modell- und Elementselektion über Konfigurationsterme

Die Elementselektion über Konfigurationsterme ist gegenüber der Elementselektion über Konfigurationsattribute dadurch charakterisiert, dass hier Modelle und Modellelemente direkt mit Konfigurationsparameterausprägungen in Beziehung gesetzt werden, anstatt sie zunächst durch die Vergabe von Eigenschaften in Kategorien einzuteilen.

Die *Konfigurationsterme*, die den *Modellen* oder *Modellelementausprägungen* annotiert werden können, enthalten direkt die Informationen, die sie als relevant für Konfigurationsparameterausprägungen kennzeichnen (vgl. hierzu und im Folgenden Abbildung 10). Da Modelle und Modellelemente auch für mehrere Konfigurationsparameterausprägungen relevant sein können, ist auch hier – analog zu den Konfigurationsattributen – eine Kombination durch logische Verknüpfung verschiedener Konfigurationsterme möglich. Konfigurationsterme werden deshalb spezialisiert in einfache Konfigurationsterme, die der *KP-KPA-Zuordnung* entsprechen sowie in *komplexe Konfigurationsterme*, die durch logische Operatoren (*AND* bzw. *OR*) verknüpfte Mengen von KP-KPA-Zuordnungen darstellen.

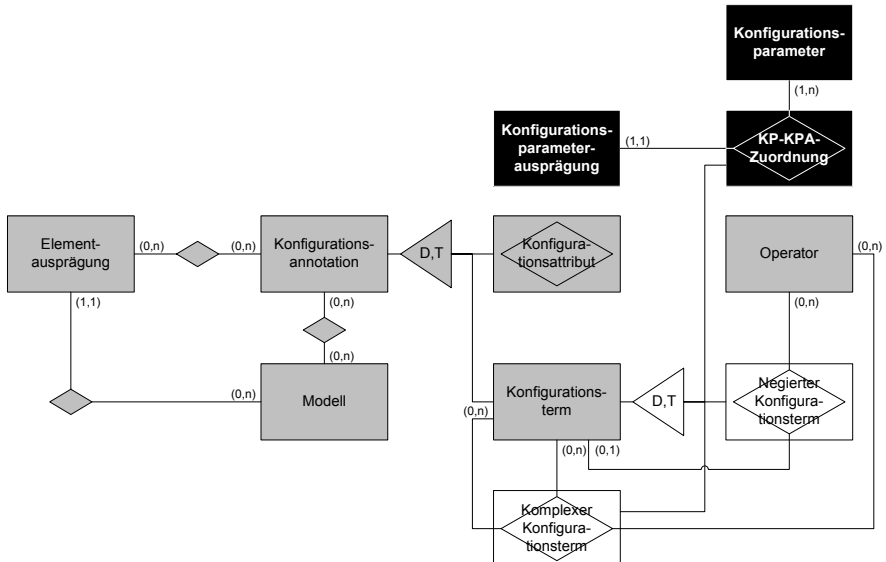


Abbildung 10: Element- und Modellselektion über Konfigurationsterme



Sollen mehr als zwei Konfigurationsterme verknüpft werden, so lassen sich über die Generalisierung vom *komplexen Konfigurationsterm* zum *Konfigurationsterm* verschachtelte logische Ausdrücke von Konfigurationstermen aufbauen. Zusätzlich kann einem Konfigurationsterm ein Operator direkt zugewiesen werden – nämlich genau dann, wenn der Term durch einen *NOT*-Operator negiert werden soll (vgl. *negierten Konfigurationsterm*). Negierte Konfigurationsterme können analog zu komplexen Konfigurationstermen über die Generalisierung zum *Konfigurationsterm* verschachtelt werden. Ein Beispiel für einen komplexen Konfigurationsterm, der wiederum mehrere komplexe Konfigurationsterme enthält, ist der folgende:

(*Geschäftsart (Lager) AND Geschäftsart (Strecke)*) OR  
(*Geschäftsart (Lager) AND Geschäftsart (Strecke)*  
*AND Geschäftsart (Zentralregulierung)*)

Er drückt aus, dass das Modellelement, dem dieser Term zugewiesen ist, nur dann angezeigt wird, wenn das Referenzmodell für ein Unternehmen konfiguriert wird, das entweder die Geschäftsarten *Lager* und *Strecke* gleichzeitig praktiziert, oder aber alle drei Geschäftsarten.

Ein Konfigurationsprozess, der durch die Elementselektion über Konfigurationsterme umgesetzt wird, läuft wie folgt ab:

- Die ausgewählte Konfigurationsparameterausprägung wird mit den Konfigurationstermen sämtlicher Modelle und Modellelemente verglichen.
- Erfüllt die Konfigurationsparameterausprägung den Konfigurationsterm, so wird das Element bzw. das Modell angezeigt. Anderenfalls wird es ausgeblendet.

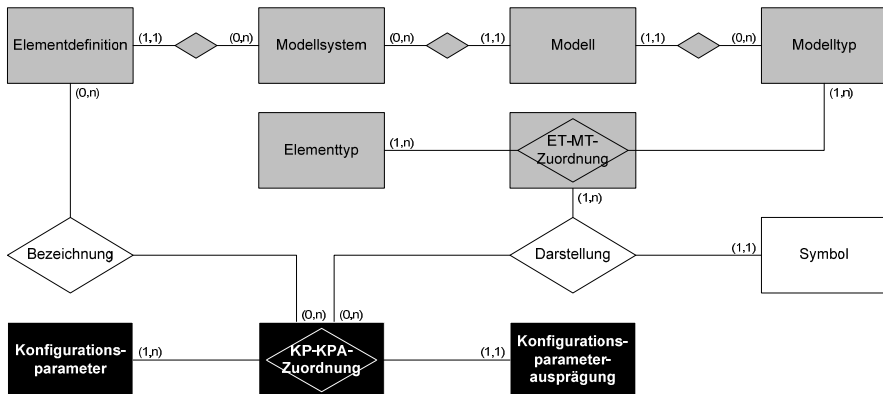
### ***Vergleich der Modell- und Elementselektion über Konfigurationsattribute sowie über Konfigurationsterme***

Trotz ihrer Äquivalenz wird keiner der beiden Konfigurationsmechanismen als redundant eingestuft. Vielmehr sind die Mechanismen für unterschiedliche Anwendungsszenarien relevant. Die Elementselektion über Konfigurationsterme eignet sich für die Konfiguration von einzelnen Modellelementen, für die keine spezielle Eigenschaft als Konfigurationsgrundlage mit Ausnahme der expliziten Zugehörigkeit zu einem konkreten Anwendungskontext identifiziert werden kann. Umgekehrt eignet sich die Elementselektion über Konfigurationsattribute für modellübergreifende Gruppen von Elementen, die gemeinsame identifizierende Eigenschaften aufweisen.

Die Benutzerführung bei der Anwendung der Konfigurationsmechanismen würde durch Kappung eines der beiden Konfigurationsmechanismen erschwert. Bei der exklusiven Unterstützung der Elementselektion über Konfigurationsattribute wäre zur Substitution der Elementselektion über Konfigurationsterme eine sehr umfassende Liste von konfigurationsbezogenen Attributen pro Modellelement bereitzustellen. Umgekehrt wäre bei exklusiv vorhandener Elementselektion über Konfigurationsterme bei jedem Modellelement explizit als Term anzugeben, für welche Konfigurationsparameterausprägungen es zur Verfügung steht. Beim Anlegen neuer Konfigurationsparameter wäre jedes Mal eine Überarbeitung sämtlicher Terme notwendig. Ferner wäre der Bezug zu der Eigenschaft, die die konfigurationsabhängige Selektion auslöst, nicht mehr erkennbar, weswegen vom Modellersteller und -anwender zusätzliche Transferleistungen zu erbringen wären. Die Bereitstellung beider Konfigurationsmechanismen begründet sich folglich in der Steigerung der Effizienz seitens des Modellerstellers bei der Entwicklung des konfigurativen Modells.

### **Bezeichnungsvariation und Darstellungsvariation**

Die Bezeichnungsvariation wird realisiert, indem Elementdefinitionen abhängig von der aktuellen Konfigurationsparameterausprägung ihre Bezeichner zugeordnet werden (vgl. hierzu und zur Darstellungsvariation Abbildung 11). Auf diese Weise ändert sich die *Bezeichnung* des jeweiligen Modellelements entsprechend.



**Abbildung 11:** Bezeichnungs- und Darstellungsvariation

Für die Realisierung der Darstellungsvariation werden Elementtypen abhängig von der aktuellen Konfigurationsparameterausprägung und abhängig von dem Modelltyp, dem sie zugewiesen sind, unterschiedliche *Sym-*

bole zugeordnet. Diese Beziehung wird durch den Relationshiptyp *Darstellung* hergestellt. Die Unterscheidung der zugeordneten Modelltypen ist notwendig, da Elementtypen bei gleicher Konfigurationsparameterausprägung in unterschiedlichen Modelltypen unterschiedliche Symbole haben können.

Die in den Abbildungen 6-11 dargestellten Datenmodelle sind Teil eines integrierten Gesamtdatenmodells, das nochmals in Abbildung 12 zusammengefasst ist. Das Gesamtdatenmodell repräsentiert hiermit die Gesamtdatenbasis des konfigurativen Modellierungswerkzeugs.

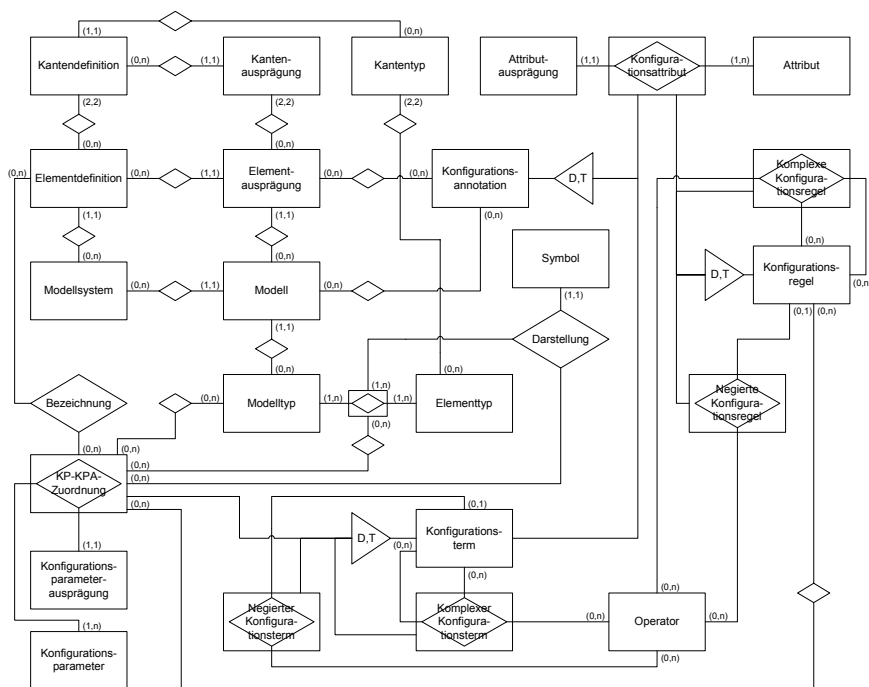


Abbildung 12: Gesamtdatenmodell

## 4 Implementierungsaspekte

Die Konfigurationsunterstützung vorhandener Werkzeuge ist zum momentanen Zeitpunkt nur bedingt gegeben. Werkzeuge mit umfangreicher Skriptsprachenfunktionalität wie *ConceptBase* oder *Cubetto* stellen viel versprechende Plattformen dar, die für den Zweck der Konfiguration erweitert werden können. Hierbei ist allerdings die Frage zu stellen, ob sich solche Werkzeuge, die sich gegen etablierte, kommerzielle Werkzeuge wie

bspw. *ARIS* behaupten müssen, durchsetzen können. Die Ergebnisse einer empirischen Studie [DeKn07] zeigen, dass sich der Bekanntheits- und damit Verbreitungsgrad solcher Werkzeuge in Grenzen hält, was diese Vermutung untermauert. Grundsätzlich vielversprechend erscheint ebenfalls der Ansatz, ein Modellierungswerkzeug auf Basis eines weit verbreiteten Zeichenprogramms zu entwickeln, wie er von *Semtalk* verfolgt wird. Hier fehlt allerdings die Möglichkeit der Spezifikation von Konfigurationsfunktionalität (vgl. nochmals [DeKn07]).

Ausgehend von diesen Überlegungen wird auf der Basis des Fachkonzepts zur Konfiguration von Informationsmodellen ein Aufsatz für das am weitesten verbreitete Werkzeug entwickelt. Der Konfigurationsaufsatz *adapt(x)* wird für das Modellierungswerkzeug *ARIS* entwickelt, der in der zum momentanen Zeitpunkt vorhandenen Version die vorgestellten Konfigurationsmechanismen unterstützt (vgl. zu einer ausführlichen Beschreibung dieses Werkzeugs die Beiträge von TOBIAS RIEKE und ARMIN STEIN sowie von PATRICK DELFMANN, TOBIAS RIEKE und ARMIN STEIN in diesem Band).

## 5 Ausblick

Mit dem Ziel, Softwareentwicklern eine möglichst umfassende methodische Unterstützung für die Verwaltung von Softwaremodellvarianten bzw. für die Anpassung von Softwaremodellen zu geben, bietet sich neben der Konfiguration die zusätzliche Berücksichtigung der in Abschnitt 1 bereits erwähnten übrigen Wiederverwendungsansätze an. Um eine möglichst einfache Verwendung der unterschiedlichen Ansätze zu gewährleisten, sind diese in einer gemeinsamen Modellierungstechnik zu integrieren. Das Fachkonzept des konfigurativen Referenzmodellierungswerkzeugs ist auf Basis dieser Technik zu erweitern. Ansätze zur Integration konfigurativer und anderer Wiederverwendungsansätze, die sich für die Erweiterung des hier vorgestellten Fachkonzepts eignen, finden sich insbesondere bei [BeDK04; Delf06].

## Literaturverzeichnis

[Allw98] Allweyer, T.: Modellbasiertes Wissensmanagement. Information Management & Consulting 13 (1998) 1, S. 37-45.

- [BDKK02] Becker, J.; Delfmann, P.; Knackstedt, K.; Kuropka, K.: Konfigurative Referenzmodellierung. In: Becker, J.; Knackstedt, R. (Hrsg.): Wissensmanagement mit Referenzmodellen. Konzepte für die Anwendungssystem- und Organisationsgestaltung. Heidelberg 2002, S. 25-144.
- [BeDK04] Becker, J.; Delfmann, P.; Knackstedt, R.: Konstruktion von Referenzmodellierungssprachen. Ein Ordnungsrahmen zur Spezifikation von Adaptionsmechanismen für Informationsmodelle. Wirtschaftsinformatik 46 (2004) 4, S. 251-264.
- [BeSc04] Becker, J.; Schütte, R.: Handelsinformationssysteme. 2. Auflage, Frankfurt am Main 2004.
- [BKKD03] Becker, J.; Knackstedt, R.; Kuropka, D.; Delfmann, P.: Konfiguration fachkonzeptioneller Referenzmodelle. In: Uhr, W.; Esswein, W.; Schoop, E. (Hrsg.): Wirtschaftsinformatik 2003, Band II. Medien – Märkte – Mobilität. Heidelberg 2003, S. 901-920.
- [Bro03] vom Brocke, J.: Referenzmodellierung. Gestaltung und Verteilung von Konstruktionsprozessen. Berlin 2003.
- [Chen76] Chen, P. P.-S.: The Entity-Relationship Model. Toward a Unified View of Data. ACM Transactions on Database-Systems 1 (1976) 1, S. 9-36.
- [DeKn07] Delfmann, P.; Knackstedt, R.: Konfiguration von Informationsmodellen. Untersuchungen zu Bedarf und Werkzeugunterstützung. In: Oberweis, A.; Weinhardt, C.; Gimpel, H.; Koschmider, A.; Pankratius, V.; Schnizler, B. (Hrsg.): eOrganisation: Service-, Prozess-, Market-Engineering. Proceedings der 8. Internationalen Tagung Wirtschaftsinformatik. Band 2. Karlsruhe 2007, S. 127-144.
- [Delf06] Delfmann, P.: Adaptive Referenzmodellierung. Methodische Konzepte zur Konstruktion und Anwendung wiederverwendungsorientierter Informationsmodelle. Berlin 2006.
- [Groc82] Grochla, E.: Grundlagen der organisatorischen Gestaltung. Stuttgart 1982.
- [Hamm97] Hamm, V.: Informationstechnikbasierte Referenzprozesse. Prozessorientierte Gestaltung des industriellen Einkaufs. Wiesbaden 1997.
- [Hamm99] Hammel, C.: Generische Spezifikation betrieblicher Anwendungssysteme. Aachen 1999.
- [HaPS99] Han, T.-D.; Purao, S.; Storey, V. C.: A Methodology for Building a Repository of Object-Oriented Design Fragments. In: Akoka, J.; Bouzeghoub, M.; Comyn-Wattiau, I.; Métais, E. (Hrsg.): Conceptual Modeling – ER '99 – 18<sup>th</sup> International Conference on Conceptual Modeling, Paris, France, November 15-18, 1999 Proceedings. Berlin et al. 1999, S. 203-217.
- [Hars94] Hars, A.: Referenzdatenmodelle. Grundlagen effizienter Datenmodellierung. Wiesbaden 1994.
- [KeNS92] Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage „Ereignisgesteuerter Prozeßketten (EPK)“. In: Scheer, A.-W. (Hrsg.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 89. Saarbrücken 1992.

- [Knac06] Knackstedt, R.: Fachkonzeptionelle Referenzmodellierung einer Managementunterstützung mit quantitativen und qualitativen Daten. Methodische Konzepte zur Konstruktion und Anwendung. Berlin 2006.
- [Krus96] Kruse, C.: Referenzmodellgestütztes Geschäftsprozessmanagement. Ein Ansatz zur prozessorientierten Gestaltung vertriebslogistischer Systeme. Wiesbaden 1996.
- [Kurb03] Kurbel, K.: Produktionsplanung und -steuerung: methodische Grundlagen von PPS-Systemen und Erweiterungen. 5. Auflage, München et al. 2003.
- [Lan97] Lang, K.: Gestaltung von Geschäftsprozessen mit Referenzprozeßbausteinen. Wiesbaden 1997.
- [LoHM02] Lohmann, M.; Hau, M.; Mertens, P.: Anforderungsanalyse auf der Basis von Unternehmensmerkmalen. In: Becker, J.; Knackstedt, R. (Hrsg.): Wissensmanagement mit Referenzmodellen. Heidelberg 2002, S. 279-289.
- [MeLo00] Mertens, P.; Lohmann, M.: Branche oder Betriebstyp als Klassifikationskriterien für die Standardsoftware der Zukunft? Erste Überlegungen, wie künftig betriebswirtschaftliche Standardsoftware entstehen könnte. In: Bodendorf, F.; Grauer, M. (Hrsg.): Verbundtagung Wirtschaftsinformatik 2000. Aachen 2000, S. 110-135.
- [Remm97] Remme, M.: Konstruktion von Geschäftsprozessen. Ein modellgestützter Ansatz durch Montage generischer Prozesspartikel. Wiesbaden 1997.
- [Roh195] Rohloff, M.: Produktionsmanagement in modularen Organisationsstrukturen. Reorganisation der Produktion und objektorientierte Informationssysteme für verteilte Planungssegmente. München, Wien 1995.
- [RoSD05] Rosemann, M.; Schwegmann, A.; Delfmann, P.: Vorbereitung der Prozessmodellierung. In: Becker, J.; Kugeler, M.; Rosemann, M. (Hrsg.): Prozessmanagement. Ein Leitfaden zur prozessorientierten Organisationsgestaltung. 5. Auflage, Berlin et al. 2005, S. 45-103.
- [Sche97] Scheer, A.-W.: Wirtschaftsinformatik. Referenzmodelle für industrielle Geschäftsprozesse. 7. Auflage, Berlin et al. 1997.
- [Schr96] Schreyögg, G.: Organisation. Grundlagen moderner Organisationsgestaltung. Wiesbaden 1996.
- [Schu01] Schulze D.: Grundlagen der wissensbasierten Konstruktion von Modellen betrieblicher Systeme. Aachen 2001.
- [Schü98] Schütte, R.: Grundsätze ordnungsmäßiger Referenzmodellierung. Konstruktion konfigurations- und anpassungsorientierter Modelle. Wiesbaden 1998.
- [Schw99] Schwegmann, A.: Objektorientierte Referenzmodellierung. Theoretische Grundlagen und praktische Anwendung. Wiesbaden 1999.
- [ScSt83] Schlageter G.; Stucky, W.: Datenbanksysteme – Konzepte und Modelle. 2. Auflage, Stuttgart 1983.
- [Spie93] Spiegel, H.: Methodik zur Analyse und Dokumentation fachlicher Begriffswelten innerhalb des Unternehmens TELEKOM. Darmstadt 1993.
- [Wolf01] Wolf, S.: Wissenschaftstheoretische und fachmethodische Grundlagen der Konstruktion von generischen Referenzmodellen betrieblicher Systeme. Aachen 2001.