

September 2003

Konfiguration fachkonzeptioneller Referenzmodelle

Jörg Becker

Westfälische Wilhelms-Universität Münster, becker@wi.uni-muenster.de

Ralf Knackstedt

Westfälische Wilhelms-Universität Münster, israkn@wi.uni-muenster.de

Dominik Kuropka

Westfälische Wilhelms-Universität Münster, isdoku@wi.uni-muenster.de

Patrick Delfmann

Westfälische Wilhelms-Universität Münster

Follow this and additional works at: <http://aisel.aisnet.org/wi2003>

Recommended Citation

Becker, Jörg; Knackstedt, Ralf; Kuropka, Dominik; and Delfmann, Patrick, "Konfiguration fachkonzeptioneller Referenzmodelle" (2003). *Wirtschaftsinformatik Proceedings 2003*. 98.

<http://aisel.aisnet.org/wi2003/98>

This material is brought to you by the Wirtschaftsinformatik at AIS Electronic Library (AISeL). It has been accepted for inclusion in Wirtschaftsinformatik Proceedings 2003 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

In: Uhr, Wolfgang, Esswein, Werner & Schoop, Eric (Hg.) 2003. *Wirtschaftsinformatik 2003: Medien - Märkte - Mobilität*, 2 Bde. Heidelberg: Physica-Verlag

ISBN: 3-7908-0111-9 (Band 1)

ISBN: 3-7908-0116-X (Band 2)

© Physica-Verlag Heidelberg 2003

Konfiguration fachkonzeptioneller Referenzmodelle

**Jörg Becker, Ralf Knackstedt,
Dominik Kuropka, Patrick Delfmann**

Westfälische Wilhelms-Universität Münster

Zusammenfassung: Bei der projektspezifischen Anpassung von fachkonzeptionellen Referenzmodellen können sämtliche Modellaspekte von Änderungen betroffen sein. Für die Formulierung vorgedachter Modellvarianten im Rahmen der konfigurativen Referenzmodellierung ist durch diese allgemeine Feststellung allerdings wenig gewonnen. Dieser Beitrag stellt mit der Definition von Konfigurationsparametern und -mechanismen konkrete Hilfsmittel vor, die den Referenzmodellierer bei der Operationalisierung seiner Aufgabenstellung unterstützen sollen. Die Konfiguration wird dabei zunächst in das Spektrum der Ansätze zur Wiederverwendung fachkonzeptioneller Informationsmodelle eingeordnet (Abschnitt 1). Anschließend werden mit Konfigurationsparametern (Abschnitt 2) und -mechanismen (Abschnitt 3) Konzepte zur Reduktion und Bewältigung der Komplexität der konfigurativen Referenzmodellierung präsentiert. Für die Spezifikation der Konfigurationsmechanismen wird ein Ordnungsrahmen vorgestellt, dessen Inhalte mit fachkonzeptioneller Schwerpunktsetzung erläutert werden. Zum Schluss des Beitrags werden verwandte Arbeiten gewürdigt (Abschnitt 4) und ein Fazit gezogen (Abschnitt 5).¹

Schlüsselworte: Referenzmodellierung, Konfiguration, Modellvariantenmanagement, Multiperspektivität

1 Konfiguration als Konstruktionsansatz für Referenzmodelle

Die Effektivität und Effizienz der fachkonzeptionellen Modellierung kann erhöht werden, indem bestehende Modelle als Ausgangslösung für die Entwicklung projektspezifischer Modelle genutzt werden [Schü98, S. 367ff.]. In einer entspre-

¹ Die hier präsentierten Erweiterungen von Modellierungstechniken sind im Rahmen des von der Deutschen Forschungsgemeinschaft (DFG) finanzierten Projektes „Konstruktion konfigurierbarer Referenzmodelle für die integrierte Anwendungssystem- und Organisationsgestaltung“ (DFG-Kennung 196495) entwickelt worden.

chenden Modellbeziehung werden als Ausgangslösung dienende Modelle als Referenzmodelle bezeichnet (zu einer engeren Begriffsfassung vgl. [Schü98, S. 69ff.]). Für die Ableitung projektspezifischer Modelle eignen sich unterschiedliche Ansätze zur Wiederverwendung fachkonzeptioneller Modelle, die auch kombiniert eingesetzt werden können [Broc02, S. 319ff.]. Der Vergleich der Ansätze zeigt, dass sie in unterschiedlichem Ausmaß eine vom Referenzmodell-ersteller zu leistende Vorwegnahme von im Rahmen der Referenzmodell-anwendung notwendig werdenden Modellanpassungen vorsehen. In vielen Fällen ist eine Anpassung des Referenzmodells durch *Spezialisierung* seiner Modellteile vorgesehen [Rohl95]. Der Detaillierungsgrad des Referenzmodells wird in diesem Fall bewusst eingeschränkt. Ist das Referenzmodell in Komponenten zerlegt, können diese Komponenten im Rahmen der *Aggregation* zu neuen Lösungen zusammengesetzt werden [Lang97; Remm97; Hamm99; HaPS99; Schu01]. Die Kombinierbarkeit kann dabei durch Schnittstellendefinitionen eingeschränkt werden. Die *Instanziierung* eines Referenzmodells sieht dagegen vor, dass Platzhalter durch zulässige Ausprägungen ihres Wertebereichs gefüllt werden, wodurch die Anpassung des Modells explizit beschränkt wird [Schü98; Schw99; Schu01; Wolf01]. Die *Konfiguration* ist ebenfalls dadurch charakterisiert, dass das Referenzmodell explizite Anpassungspunkte enthält. In diesem Fall sind zu den Anpassungspunkten Regeln formuliert, die festlegen, wie das Referenzmodell in Abhängigkeit von aktuellen Konfigurationsparameterausprägungen anzupassen ist [Schü98; Schw99; Schl00].

Die Konfiguration eignet sich in besonderem Maße für die Formulierung von konkreten Gestaltungsempfehlungen und für die Anpassung des Referenzmodells an vorausgeplante Anwendungskontexte. Das Problem, dass in der Regel nicht sämtliche Anpassungsbedarfe vorweggenommen werden können, lässt sich leicht dadurch reduzieren, dass im Anschluss an die Konfiguration eine kreative Anpassung des konfigurierten Modells zugelassen wird, wobei die übrigen genannten Anpassungsmechanismen zum Einsatz kommen können. Motiviert durch das Potenzial der Konfiguration einen wertvollen Beitrag zur Verwaltung vorgedachter Modellvarianten und damit für die Bereitstellung passgenauerer Ausgangsmodelle für unterschiedliche Anwendungskontexte leisten zu können, werden im Folgenden Erweiterungen von Modellierungstechniken vorgeschlagen, die eine konfigurative Referenzmodellierung umfassend unterstützen.

Die Vorwegnahme von Modellanpassungen im Rahmen der Modellierung konfigurierbarer Referenzmodelle ist durch ein Komplexitätsproblem geprägt, das darin besteht, dass prinzipiell jeglicher Aspekt des Modells von einer Anpassungsmaßnahme betroffen sein kann. „Each information engineer uses its own rules, heuristics, graphical representation and so on. [...] Summarizing, it is clear that unrestricted, adequate, flexible support of information modelling is, practically speaking, impossible to achieve.“ [HoVe96, S. 65-66] Der Referenzmodell-ersteller muss die berücksichtigten Anpassungsbedarfe daher vor dem Hintergrund von Nutzenabwägungen systematisch einschränken. Zur Reduktion und Beherrschung

der Komplexität [Rose96a; Blis00] der Konfiguration greift der vorgestellte Modellierungsansatz auf Konfigurationsparameter und -mechanismen zurück.

2 Konfigurationsparameter

Ein Aspekt der Komplexität der Konfiguration betrifft die Parameter, an die das Modell angepasst wird. Die Konfigurationsparameterausprägungen sollen den Anwendungskontext des Referenzmodells beschreiben. Der hier vorgestellte Ansatz beschreibt den Anwendungskontext über Unternehmensmerkmale und Perspektiven (vgl. Abbildung 1):

- *Unternehmensmerkmale*, wie z. B. die betriebenen Geschäftsarten, dienen der Abgrenzung von Unternehmensklassen, denen Modellvarianten zugeordnet werden können (zu Unternehmenstopologien vgl. [MBGH96; MeLo00; LoHM02]). Aus der Angabe, dass ein Unternehmen die Geschäftsart Lagergeschäft nicht betreibt, kann z. B. abgeleitet werden, dass alle Prozesse, Daten, Organisationseinheiten etc., die auf das Ein-, Aus- oder Umlagern von Artikeln zurückzuführen sind, aus dem Unternehmensmodell entfernt werden können.
- *Perspektiven* dienen der Repräsentation der unterschiedlichen Anforderungen an das Referenzmodell durch Benutzerklassen.² Die Anwendungszwecke von Benutzern (z. B. Anwendungssystemgestaltung oder Organisationsgestaltung), ihre Rollen innerhalb eines Projektes (z. B. Projektmanager, Modellierer, Fachanwender) sowie weitere individuelle Einflüsse (z. B. Zuständigkeit für bestimmte Funktionsbereiche, Kenntnisse bestimmter Modellierungstechniken) können bei der Bildung von Benutzerklassen berücksichtigt werden [Rose98b, S. 11-19]. Über die Konfigurationsregeln lassen sich den Perspektiven auf sie abgestimmte Varianten des Referenzmodells zuordnen. Eine

² Wahrnehmung und Erkenntnis werden einer konstruktivistischen, erkenntnistheoretischen Position folgend als subjektabhängige Phänomene aufgefasst [Glas00, S. 29f.]. Das Subjekt richtet sein Handeln nach seiner individuell konstruierten Vorstellung von Wirklichkeit aus. Diese Subjektivierung stellt nach LUHMANN eine wesentliche Strategie zur Absorption von Komplexität dar und ist damit pragmatisch bedingt [Luhm99, S. 181ff.]. Die Subjektivierung erhöht die Lösungschancen von Problemen, indem sie ihnen eine spezielle, lösbar erscheinende Fassung verleiht. Die Qualität eines Modells wird daher von Seiten des Modellnutzers umso höher bewertet werden, desto stärker es seiner problembedingten individuellen Subjektivierung entspricht [Woll86, S. 13; Rose98b, S. 11-19]. Die Erstellung von Modellen bzw. Modellsystemen, die zugleich unterschiedlichen Subjektivierungen gerecht zu werden versuchen, wird multiperspektivische Modellierung genannt [DaSh96; Rose96b, Rose98a]. Mit der Unterscheidung verschiedener Perspektiven versuchen multiperspektivische Modelle die Differenzen zwischen den Subjektivierungen von Modellnutzern bzw. von Gruppen von Modellnutzern zu repräsentieren.

Variante für Anwendungssystemgestalter kann z. B. die Eliminierung derjenigen Informationsobjekte vorsehen, die ausschließlich für Organisationsgestalter von Relevanz sind (z. B. rein manuell durchzuführende Tätigkeiten in Prozessmodellen).

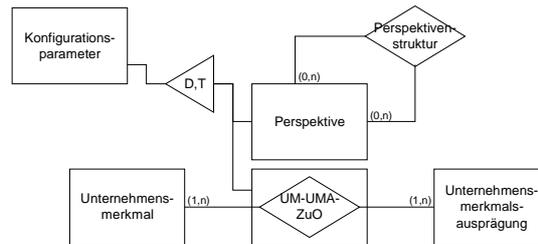


Abbildung 1: Konfigurationsparameter

Durch die Unternehmensmerkmalsausprägungen bzw. Perspektiven werden einzelne Unternehmen bzw. Modellbenutzer zu Gruppen zusammengefasst. Damit verbunden ist eine Abstraktion von nicht berücksichtigten Unterschieden zwischen den einzelnen Mitgliedern der Klassen, da die Abgrenzung der Klassen nur ausgewählte Aspekte in bestimmter Detailliertheit ihrer Ausprägungen berücksichtigt (z. B. ausschließlich die Anwendungszwecke Organisationsgestaltung und Anwendungssystemgestaltung bei Vernachlässigung von Rollen und sonstigen Einflüssen). Die Komplexität der Erstellung eines konfigurierbaren Referenzmodells kann weiter reduziert werden, indem einzelne Konfigurationsparameter eliminiert bzw. zusammengefasst werden. Diese Entscheidungen sind mit einem Prognoseproblem verbunden, da in der Regel bei der Festlegung der Parameter und Parameterausprägungen der mit ihrer Berücksichtigung verbundene Modellierungsaufwand nur unzureichend bekannt ist.

3 Konfigurationsmechanismen

3.1 Ordnungsrahmen

Um die Anwendung der Konfigurationsmechanismen für den Referenzmodell-ersteller effizient zu gestalten, ist es vorteilhaft, Mechanismen mit unterschiedlich weitem Wirkungsgrad bereitzustellen. Deshalb werden Konfigurationsmechanismen nicht allein für die Ebene der Anwendung der Modellierungstechniken vorgesehen (Modellebene), sondern auch für die Ebene ihrer Definition (der Metamodellebene). Konfigurative Anpassungen der Metamodelle wirken sich auf alle Modelle aus, die unter Rückgriff auf die jeweilige Modellierungstechnik entstanden sind. Auf Metamodellebene kann z. B. festgelegt werden, dass der

Perspektive Anwendungssystemgestaltung erweiterte Ereignisgesteuerte Prozessketten zugeordnet sind (vgl. Mechanismus (1) Modelltypselektion in Abbildung 2), wobei allerdings Organisationseinheiten nicht als Prozessressourcen Verwendung finden sollen (vgl. Mechanismus (2) Elementtypselektion in Abbildung 2). Ergänzend dazu erlauben Konfigurationsregeln auf Modellebene z. B. die perspektivenspezifische Eliminierung einzelner Prozessfunktionen (vgl. Mechanismus (3b) Elementselektion über Attribute in Abbildung 2).

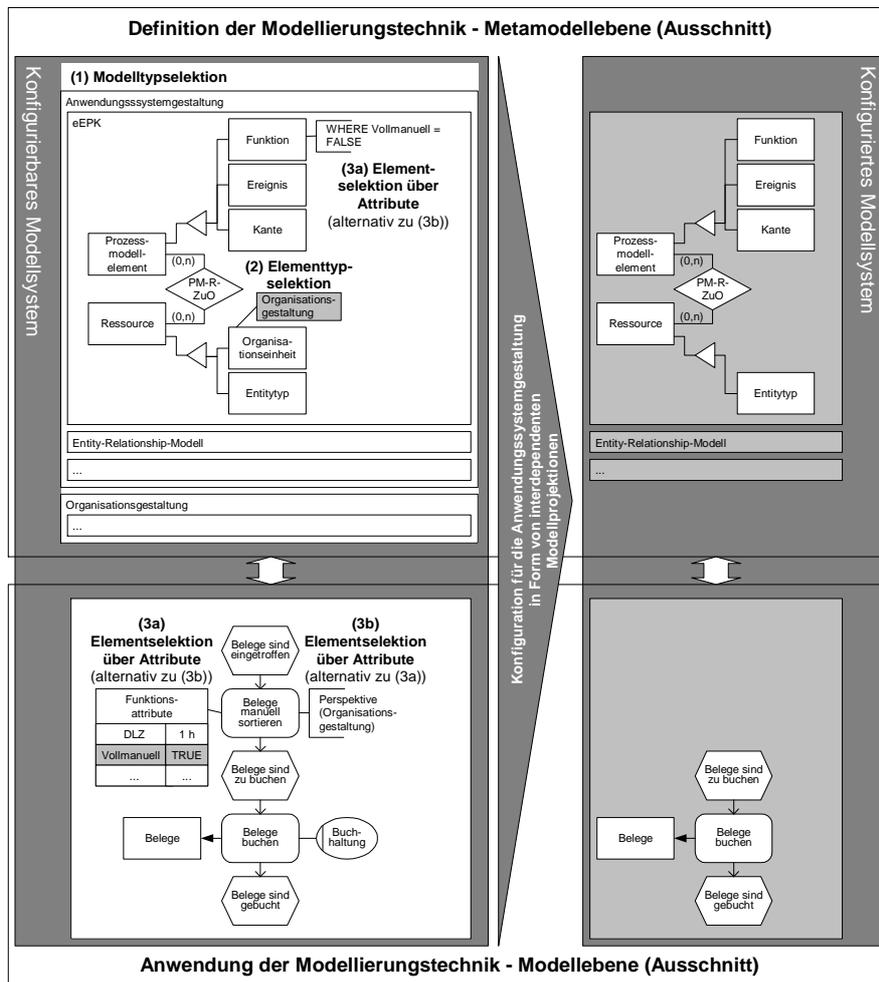


Abbildung 2: Konfigurationsmechanismen

Zur Strukturierung der Spezifikation der vorgenommenen Spracherweiterungen wird auf einen Ordnungsrahmen [Meis01] zurückgegriffen, der von drei Dimensionen aufgespannt wird (vgl. Abbildung 3) [BDKK02]:

- Die erste Dimension unterscheidet fünf Kategorien von Konfigurationsmechanismen, denen zum Teil mehrere Einzelmechanismen zugeordnet sind.
- Die zweite Dimension gliedert sich in drei Modellebenen, wobei die Modelle der jeweils übergeordneten Modellebene die Sprache der Modelle der untergeordneten Ebene beschreiben [Stra96; Holt01; Ortn97]. Die Spezifikation der Mechanismen umfasst neben der Modell- und der Metamodellebene auch die Meta-Metamodellebene, um auch Spracherweiterungen, die auf der Metamodellebene z. B. zur Unterstützung der Variantenbildung zu Modelltypen notwendig sind, dokumentieren zu können.
- Ergänzt wird der Ordnungsrahmen durch die Dimension der Entwicklungsphasen der Anwendungssystemgestaltung, die sich in Anlehnung an die Beschreibungsebenen der Architektur integrierter Informationssysteme (ARIS) in das Fach- und DV-Konzept sowie die Implementierung gliedern [Sche01].

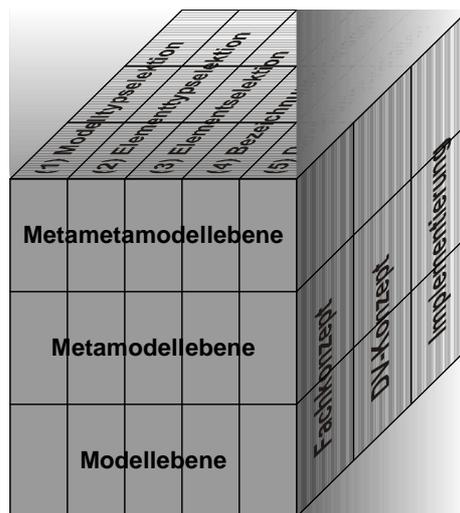


Abbildung 3: Ordnungsrahmen

Im Folgenden werden ausgewählte Segmente des Ordnungsrahmens exemplarisch und soweit für die Nachvollziehbarkeit zuträglich modellierungstechnikspezifisch konkretisiert. Dabei wird der Schwerpunkt auf die fachkonzeptionelle Ebene des Ordnungsrahmens gelegt. Der Fokus auf das Fachkonzept erlaubt die Reduktion des Ordnungsrahmens auf einen zweidimensionalen Navigator, indem die Dimension der Entwicklungsphasen ausgeblendet wird. Dieser Navigator wird im Folgenden zur Einordnung der vorgestellten Modellausschnitte verwendet.³

³ Für die Metamodellierung werden Entity-Relationship-Modelle [Chen76] mit (min, max)-Kardinalitäten [ScSt83, S. 50f.] verwendet. Repräsentationen werden exemplarisch eingeführt. Ergänzende Aspekte zur Sprachdokumentation wie Lexika [Ortn97, S.

3.2 Modelltypselektion

Die Modelltypselektion berücksichtigt die perspektivenspezifische Relevanz von Modelltypen [Rose98b, S. 11-19]. Modelltypen repräsentieren die Ergebnistypen spezieller Modellierungstechniken, die im Rahmen der Referenzmodellierungstechnik kombiniert werden können, wie zum Beispiel die der erweiterten Ereignisgesteuerten Prozessketten (eEPK) [KeNS92], Entity-Relationship-Modelle (ERM) [Chen76], Fachbegriffsmodelle [Rose98b, S. 11-19] oder Organigramme [Groc82, S. 305; Kuge00, S. 134f.]. *Teilmetamodelle* beschreiben die einzelnen Modelltypen. *Modelltypvarianten* liefern die Basis, um mittels anderer Mechanismen (vgl. Abschnitt 3.2 sowie Abbildung 4) Modelltypen modifizieren zu können. Ein Modelltyp enthält das Maximum aller möglichen erlaubten Elementtypen, welche in einer Modelltypvariante beschränkt werden können. Pro Modelltyp existiert grundsätzlich eine Standardvariante. Der Konfigurationsmechanismus ordnet *Perspektiven* den für sie relevanten Modelltypen zu und unterstützt damit eine grobgranulare Konfiguration des Modellsystems (vgl. auch Mechanismus (1) in Abbildung 2). Technisch werden aus Sicht des Modellanwenders die nicht für seine Perspektive relevanten Modelltypen ausgeblendet. Bspw. kann vorgesehen werden, dass Anwendungssystemgestaltern im Gegensatz zu Organisationsgestaltern keine Fachbegriffsmodelle, sondern stattdessen Entity-Relationship-Modelle zur Verfügung gestellt werden.

3.3 Elementtypselektion

Feingranulare Konfigurationsregeln auf Metamodellebene lassen sich mit der Elementtypselektion vornehmen. Modelltypen werden durch die ihnen zugeordneten Elementtypen und deren Beziehungen untereinander charakterisiert. In der eEPK sind dies z. B. Funktionen, Ereignisse, Kanten und Ressourcen. Der Konfigurationsmechanismus erlaubt es, zu Modelltypen Varianten zu bilden, indem Elementtypen den Perspektiven zugeordnet und ggf. ausgeblendet werden. Bspw. unterscheiden sich Varianten der eEPK in den an Funktionen annotierbaren Modellelementtypen. Kandidaten für die Annotation sind z. B. Entitytypen, Fachbegriffe, Anwendungssysteme und Organisationseinheiten. Die Modelltypvarianten werden gebildet, indem ihnen auszublendende Elementtypen zugeordnet werden. Die Perspektive Organisationsgestaltung ermöglicht z. B. die Annotation von Fachbegriffen an Funktionen, während diese für die Perspektive Anwendungssystemgestaltung durch entsprechende Datentypen/Entitytypen oder Datenmodelle ersetzt werden (vgl. Mechanismus (2) in Abbildung 2). Ein Elementtyp kann mehreren Modelltypen zugeordnet sein, um die Wiederverwendbarkeit von Elementtypen und somit auch einzelnen Elementen modellierungsspra-

95ff.], Gegenstandseinteilungen [Ortn97, S. 84f.], Metaphern [FeSi01] und Prozesse [Stra96, S. 24ff.; Holt01, S. 300f.] bleiben unberücksichtigt.

chenübergreifend zu gestatten. Elementtypen, die tatsächlich mehreren Modelltypen zugeordnet sind, stehen nach Ausblendung eines Modelltyps bei einer Modelltypselektion in anderen Modelltypen immer noch zur Verfügung. Ein *Teilmetamodell* enthält immer die für seinen Modelltyp maximal verfügbare Anzahl an Elementtypen. Variationen der Elementtypvielfalt in Modelltypen werden durch die so genannte *Modelltypvariante* repräsentiert; die jeweils pro Modelltypvariante auszublenenden Elementtypen werden in der *MTV-MME-Beschränkung* vermerkt. Jede Modelltypvariante wird einer oder mehreren *Perspektiven* zugeordnet.

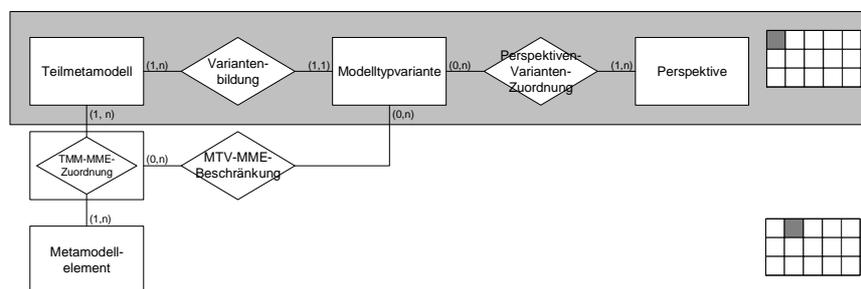


Abbildung 4: Meta-Metamodellkonstrukte zur Elementtypselektion

Da das Meta-Metamodell die Sprache des Metamodells spezifiziert, stellen Instanzen des Entitytyps *Metamodellelement* benannte Typen als Sprachelemente auf Metamodellebene dar. Werden genannte Instanzen durch Anlegen entsprechender Instanzen des Relationshiptyps *MTV-MME-Beschränkung* entfernt, sind die korrespondierenden Metamodellelemente nicht mehr Bestandteil des Metamodells. Gleiches gilt für die Modelle: Modellelemente die Instanzen einer ausgeblendeten Metamodellelementinstanz sind, werden aus den Modellen ausgeblendet.

3.4 Elementselektion

Die Konfigurationsmechanismen der Modelltyp- und Elementtypselektion ermöglichen eine perspektivenspezifische Auswahl von Elementtypen (also Elementen auf Metamodellebene). Im Gegensatz dazu wird mit der Elementselektion die Auswahl von Elementen der Modellebene verfolgt. Für die Elementselektion werden vier verschiedene Selektionskriterien vorgeschlagen, die sich in der Art der Definition der gewünschten Elementuntermengen unterscheiden:

1. Elementselektion über Typen: Sowohl Modellelemente als auch deren Beziehungen können im Rahmen der Modellierung typisiert werden (z. B. kann eine Beziehung zwischen einer Funktion und einer Organisationseinheit dem Typ „führt aus“ oder „wirkt mit bei“ angehören [Rose98b, S. 11-19]). Der Konfigurationsmechanismus der Elementselektion über Typen ermöglicht die Abgrenzung einer Untermenge der Beziehungen zwischen Modellelementen über Bezugnahme auf solche Typen. Im Rahmen der Betrachtung organisatori-

scher Zusammenhänge stellen z. B. „ist übergeordnet“, „kommuniziert mit“, „ist verantwortlich für“ und „führt aus“ mögliche, perspektivenspezifisch zuzuordnende Typausprägungen dar. Entsprechend der Zuordnung dieser Typen zur Perspektive Organisationsgestaltung kann z. B. über die Ein- bzw. Ausblendung einzelner Referenzmodellelemente entschieden werden.

2. *Elementselektion mittels Hierarchiestufenaggregation*: Modelle können hierarchische Strukturen von Modellelementen und Beziehungstypen enthalten, z. B. die disziplinarische Hierarchie einer Organisation oder hierarchisch gegliederte Kommunikationsbeziehungen; bspw. kann der Beziehungstyp „kommuniziert mit“ spezialisiert werden in „kommuniziert unilateral“ und „kommuniziert bilateral“. Um unterschiedliche, perspektivengerechte Detaillierungsgrade unterstützen zu können, ermöglicht der Konfigurationsmechanismus der Elementselektion mittels Hierarchiestufenaggregation die perspektivenspezifische Festlegung von maximalen Hierarchiestufen und die Aggregation von tiefer in der Hierarchie liegenden Elementen auf die als maximal konfigurierte Hierarchiestufe. Der Mechanismus nimmt unter Auswertung der Baumstrukturen den entsprechenden Austausch der Elemente im eigentlichen Referenzmodell vor, z. B. wird eine „Stelle X1“ der „Abteilung X“ bei definierter, maximaler Hierarchiestufe der Abteilung fortan in der entsprechenden Perspektive nicht als „Stelle X1“, sondern als „Abteilung X“ angezeigt.
3. *Elementselektion über Attribute*: Die Variantenbildung mit Hilfe der Elementselektion über Attribute erfolgt über die Auswertung von Eigenschaften, die den Referenzmodellelementen zugeordnet sind. Bspw. kann für die Funktionen einer eEPK ein Attribut zur Kennzeichnung ihres Automatisierungsgrades vorgesehen werden. Während vollautomatisierte Funktionen im besonderen Fokus der Anwendungssystemgestaltung liegen, spielen diese für die Organisationsgestaltung in der Regel keine Rolle, ebenso wie vollständig manuell zu bearbeitende Funktionen außerhalb des Interessenbereichs der Anwendungssystemgestaltung liegen (vgl. Mechanismus (3a) in Abbildung 2).
4. *Elementselektion nach Termen*: Die Abgrenzung von Elementen auf Modellebene über die perspektivenspezifische Selektion von Attributausprägungen stößt an ihre Grenzen, wenn für detaillierte Konfigurationsmaßnahmen jeweils neue Attribute eingeführt werden und damit die Menge an Attributen unübersichtlich zu werden droht. Es wird deshalb zusätzlich der Konfigurationsmechanismus der Elementselektion nach Termen eingeführt, der die direkte Zuweisung von Modellelementen zu Konfigurationsparameterausprägungen über einen booleschen Term ermöglicht. Das Konzept dieses Mechanismus' greift Vorschläge für Repräsentationsformen der unternehmensmerkmalsbasierten Modellkonfiguration (wie z. B. Build-Time-Operatoren [Schü98, S. 247]) auf, denen bisher eine metamodellbasierte Spezifizierung fehlte, identifiziert das zugrunde liegende Sprachkonzept und separiert es von Fragen der Darstellung. Mittels des einem Modellelement zugewiesenen Ausdrucks z. B. „Perspektive (Organisationsgestaltung)“ wird festgelegt, dass das entsprechende Modell-

element nicht zur Perspektive Anwendungssystemgestaltung gehört (vgl. Mechanismus (3b) in Abbildung 2). Ein Rückgriff auf die Elementselektion über Attribute ist alternativ zur Elementselektion über Terme dann zu empfehlen, wenn bei der Definition der Regeln auf bereits betrachtete Elementeigenschaften zurückgegriffen werden kann.

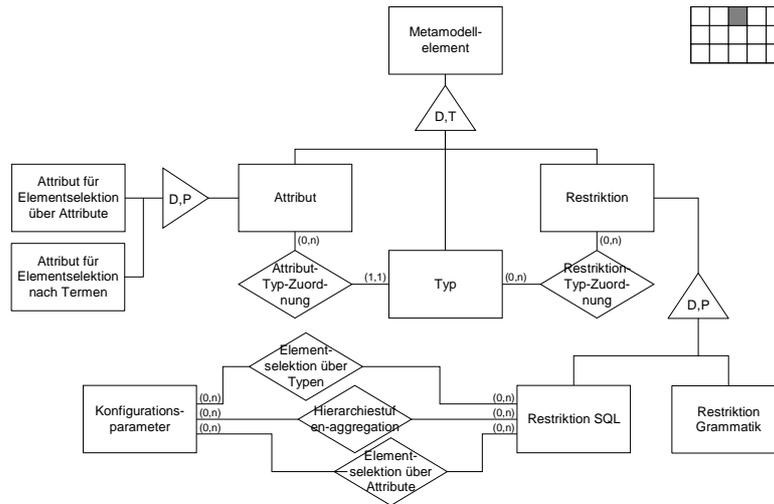


Abbildung 5: Meta-Metamodellkonstrukte zur Elementselektion

Im Folgenden werden die Elementselektionen über Typen, mittels Hierarchiestufenaggregation und über Attribute gemeinsam erklärt, da diese auf einem gemeinsamen Konzept basieren. Sie operieren auf den Ausprägungen der Metamodell-elemente. Zur Repräsentation der Elementselektionen wird auf Meta-Metamodell-ebene die Existenz von Instanzenrestriktionen für Typen auf Metamodellebene definiert (vgl. Entitytyp *Restriktion SQL* in Abbildung 5). Eine solche Restriktion ist abhängig von der jeweiligen Konfiguration, weswegen sie im Meta-Metamodell über die drei Relationstypen *Elementselektion über Typen*, *Hierarchiestufen-aggregation* und *Elementselektion über Attribute* dem Entitytyp *Konfigurationsparameter* zugeordnet ist. Die Sprache der Restriktion lehnt sich an die Standard Query Language (SQL) zur Administration, Manipulation und Abfrage von relationalen Datenbanken an [DaDa97]. Für die Elementselektion über Attribute wird im Meta-Metamodell zusätzlich ein Entitytyp *Attribut für Elementselektion über Attribute* als Spezialfall des *Attribut* Entitytyps definiert. Diese Spezialisierung ermöglicht es in der Definition einer Modellierungssprache auf Metamodellebene festzulegen, ob ein Attribut für die Elementselektion zugelassen ist. Als Konsequenz der Beschränkung der Typen auf Metamodellebene werden auf *Modellebene* alle Elemente und deren Beziehungen, die einem nicht selektierten Typ angehören, ausgeblendet.

Die Relevanz eines Modellelements für eine ausgewählte Konfiguration hängt bei der Elementselektion nach Termen von einem dem Element zugeordneten Term ab. Terme sind hierbei Attribute in Textform, deren Ausprägungen beschreiben, in welchen Konfigurationen das jeweilige Modellelement zur Verfügung steht. Terme haben zu Auswertungszwecken einer vordefinierten Grammatik (vgl. Abbildung 6) zu gehorchen.

<Term>	::= <Ausdruck> {<Operator> <Ausdruck>}	
<Ausdruck>	::= <Präfix> "Perspektive" <Perspektiven-Ausprägungsliste>	
<Ausdruck>	::= <Präfix> <Unternehmensmerkmal> <Unternehmensmerkmals-Ausprägungsliste>	
<Perspektiven-Ausprägungsliste>	::= "(" <Präfix> <Perspektiven-Ausprägungsliste> { <Operator> <Präfix> <Perspektiven-Ausprägungsliste> }")"	
<Perspektiven-Ausprägungsliste>	::= <Perspektive>	
<Unternehmensmerkmals-Ausprägungsliste>	::= "(" <Präfix> <Unternehmensmerkmals-Ausprägungsliste> { <Operator> <Präfix> <Unternehmensmerkmals-Ausprägungsliste> }")"	
<Unternehmensmerkmals-Ausprägungsliste>	::= <Unternehmensmerkmalsausprägung>	
<Operator>	::= " " "+"	
<Präfix>	::= "NOT" <leer>	

Abbildung 6: Auszug aus der Grammatik zur Elementselektion nach Termen.⁴

Realisiert wird die Elementselektion nach Termen indem im Meta-Metamodell ein spezieller Attributtyp (vgl. Entitytyp *Attribut* und dessen Spezialisierung *Attribut für Elementselektion nach Termen* in Abbildung 5) für Typen auf Metamodellenebene eingeführt wird. Um die formale Korrektheit der Terme sicherzustellen, muss die Existenz einer *Restriktion* definiert werden, die die Formulierung von Termen innerhalb von Attributen dieses Typs in der vorgestellten Grammatik sichert (vgl. die Spezialisierung *Restriktion Grammatik* von Entitytyp *Restriktion*).

Die im Meta-Metamodell definierte Sprachkomponente macht sich auch auf *Metamodellenebene* im speziell für die Elementselektion nach Termen angelegten Teilmetamodell bemerkbar (vgl. Abbildung 7). Hier werden alle Metamodellelemente aus allen Modellierungssprachen, für die eine Termzuweisung möglich ist, zu einem Metamodellelement generalisiert (vgl. Abbildung 7, in der aus Gründen der Übersichtlichkeit nur ein Teil aller für eine Termzuweisung geeigneten Modellelemente aufgeführt sind). Ein *für eine Termzuweisung geeigneter Elementtyp* steht mit dem *booleschen Konfigurationsparameterterm* in Beziehung. Dieser hat wiederum ein Attribut, welches den *Termtext* enthält und das eben dem auf Meta-Metamodellebene definierten Typ (*Attribut für Elementselektion nach Termen*) angehört (weshalb es mit einem „&“ gekennzeichnet ist). Die Restriktion, die an den Term annotiert ist, ist vom Typ *Restriktion Grammatik* und sichert die Formulierung des Termtextes in der bereits vorgestellten Grammatik). Die auf Meta-Metamodellebene definierte *Restriktion Grammatik* besitzt also *genau eine* Instanz auf Metamodellebene, nämlich diejenige, die die Formulierung von Termen in der

⁴ Auf die Darstellung trivialer Elemente der Grammatik (wie z. B. die Definition des gültigen Alphabetes und die Einschränkung von <Perspektive> und <Unternehmensmerkmalsausprägung> auf gültige Elemente) ist hier verzichtet worden.

vorgestellten Grammatik fordert. Im vorgestellten Fall handelt es sich um die erweiterte Backus-Naur-Form (EBNF) [Loud94, S. 83ff.].

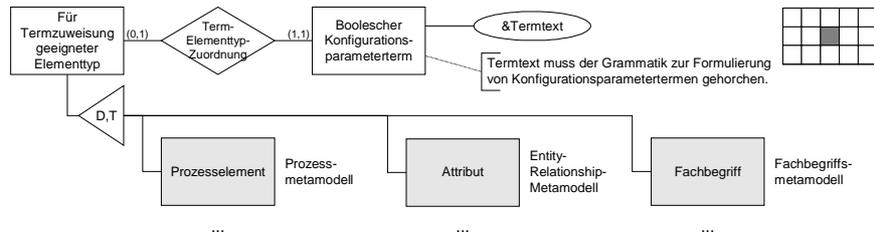


Abbildung 7: Metamodellkonstrukte zur Elementselektion nach Termen

Das Ergebnis für die Modellebene ist, dass jedem Modellelement, das dem Typ *für Termzuweisung geeigneter Elementtyp* angehört, textuell Terme hinterlegt werden können, die von einem entsprechenden Interpreter ausgewertet werden können. Anschließend kann eine Ausblendung gemäß der im Term definierten Konfigurationsparameterausprägungen durchgeführt werden.

3.5 Bezeichnungsvariation

Je nach Perspektive können in Referenzmodellen verschiedene Begriffskonventionen gelten. Je nach Benutzergruppe wird bspw. von Rechnung oder von Faktura bzw. von Auftrag oder Bestellung gesprochen, wobei mit den verschiedenen Begriffen jeweils die gleichen Sachverhalte angesprochen werden. Der Konfigurationsmechanismus der Bezeichnungsvariation ermöglicht einen von der Konfigurationsparameterausprägung abhängigen Austausch der Begriffe. Für eine solche Begriffsmenge gleicher Bedeutung (Synonyme) wird bei der Nutzung im konfigurierbaren Referenzmodell ein entsprechender Platzhalter verwendet, an dessen Stelle nach der Konfiguration der Konfigurationsparameterausprägung der zugeordnete Begriff tritt.

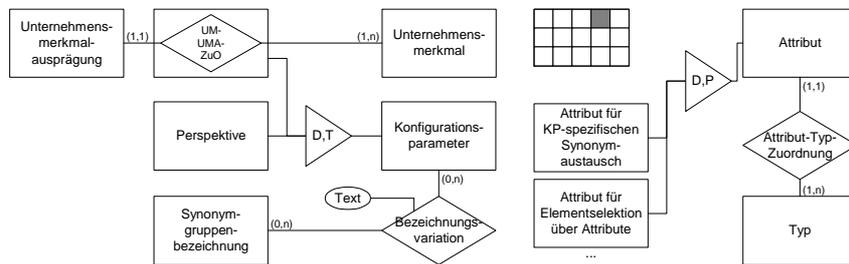


Abbildung 8: Meta-Metamodellkonstrukte zur Bezeichnungsvariation

Im Meta-Metamodell stehen zur Realisierung der Bezeichnungsvariation *Konfigurationsparameter* in Beziehung zu *Synonymgruppenbezeichnungen* (vgl. Abbildung 8) und repräsentieren somit über diese Beziehung die Umbenennungstabelle für die Modellelemente. Diese Tabelle ist modellübergreifend (daher auf Metamodellebene) zu spezifizieren. Synonymgruppenbezeichnungen definieren den führenden Standardbegriff (z. B. „Faktura“), der in Abhängigkeit von der Konfigurationsparameterausprägung durch den im Attribut *Text* gespeicherten Term (z. B. „Rechnung“) ersetzt wird.

Um eine perspektivenspezifische Umbenennungstabelle realisieren zu können, ist es notwendig, die Sprache der Metamodellebene zu erweitern. Modellelementbenennungen sind auf der Metamodellebene Attribute von Modellelementen. Da die Umbenennungstabelle nicht pauschal für alle Attribute der Modellelemente gelten soll, müssen diejenigen Attribute gekennzeichnet werden, auf die die Umbenennungstabelle anzuwenden ist. Bspw. kann dies durch ein dem Attributnamen vorangesetztes Sternchen (*) geschehen. Diese Spracherweiterung der Metamodellebene muss konsequenterweise auf der Meta-Metamodellebene durch eine Spezialisierung des Entitytyps *Attribut* in *Attribut für KP-spezifischen Synonymaustausch* modelliert werden.

3.6 Darstellungsvariation

Während mit den bisher vorgestellten Konfigurationsmechanismen die konzeptionellen Sprachaspekte der Referenzmodelle modifiziert werden können, wird mit dem Konfigurationsmechanismus der Darstellungsvariation eine Variation der repräsentationellen Sprachaspekte ermöglicht. Dabei werden die folgenden drei Typen der Darstellungsvariation unterschieden:

1. *Darstellungsvariation der Symbole*: Je nach Perspektive kann eine differierende Visualisierung von Modellelementtypen gewünscht sein. Durch den Konfigurationsmechanismus der Darstellungsvariation können dem gleichen konzeptionellen Sprachaspekt unterschiedliche Symbole zugeordnet werden. Bspw. bietet es sich an, für anwenderorientierte Benutzergruppen die klassischen Symbole der EPK durch anschaulichere Bildzeichen auszutauschen, z. B. Schreibtische anstelle von abgerundeten Rechtecken für Funktionen [Allw98].
2. *Darstellungsvariation der Topologie*: Zusätzlich zu Symbolvariationen sind häufig Variationen in der topologischen Anordnung der Modelle zu beobachten: Bspw. unterscheiden sich Benutzer in den von ihnen präferierten Leserichtungen. Eine von der Standardnotation abweichende Darstellung der eEPK ist die Spaltendarstellung, bei der die Annotation von Ressourcen an Prozessfunktionen in eigens dafür vorgesehenen Spalten erfolgt [Rose98b, S. 11-19]). Diese stellt ein Beispiel für den Einsatz dieses Mechanismus dar.

3. *Darstellungsvariation der Konfigurationsansatzpunkte*: Bisher wurde die Formulierung von Konfigurationsregeln konzeptionell betrachtet. Die Verwendung unterschiedlicher Ansätze zur Repräsentation der Konfigurationsregeln wird durch Darstellungsvarianten der Konfigurationsansatzpunkte berücksichtigt. Unter Konfigurationsansatzpunkten werden Stellen in Modellen verstanden, die Bereiche markieren, in denen sich durch die Konfiguration Änderungen ergeben. Bei der Darstellung des konfigurierbaren – noch nicht konfigurierten – Modells werden Konfigurationsansatzpunkte bspw. alternativ als sog. Build-Time-Operatoren [Schü98, S. 235-291], durch Annotation von Selektionskriterien (vgl. Abbildung 2) oder alternativ in Spaltendarstellung visualisiert.

4 Verwandte Arbeiten

Heutige CASE-Tools bzw. Upper-CASE-Tools bieten i. d. R. ein festes Kontingent an Modellierungstechniken. Einzelne Konfigurationsmechanismen sind in kommerziellen Werkzeugen bereits vorhanden. Bspw. stellt das *ARIS Toolset* [Sche96] zur Differenzierung von Modellsystemen über unterschiedliche Modellierungstechniken Methodenfilter zur Verfügung. Alternative Layoutkonventionen können über Modellvarianten berücksichtigt werden. Die automatisierte Erstellung perspektivenspezifischer Modelle lässt sich über die Programmierung von Skripten realisieren. Skripte können auch eingesetzt werden, um die Konsistenz zwischen einzelnen Modellen sicherzustellen. Eine umfassende und benutzerfreundliche Umsetzung der hier genannten Konfigurationsmechanismen steht für das kommerzielle Umfeld noch aus.

Im Unterschied zu CASE-Tools ermöglichen Meta-CASE-Tools die Erstellung von Metamodellen. Beispiele für Meta-CASE Tools sind das kommerzielle Meta-CASE-Tool *Adonis* [JKSK00] und der Forschungsprototyp *Generischer Modelleditor* (GME) [GrEs00; GeME03]. Die Unterstützung der Metamodellenebene hebt die den CASE-Tools eigene Beschränkung auf vordefinierte Modellierungssprachen auf. Häufig verwendete Meta-Modellierungssprachen sind die Unified-Modeling-Language und an ERM-Notationen angelehnte Dialekte wie EARA/GE [ThMS03]. Die Konfigurierbarkeit von Modellen und Meta-Modellen ist bei Meta-CASE-Tools allerdings wenig verbreitet, was sich im Fehlen der konfigurationsbezogenen Spracherweiterungen zeigt. Der vorgestellte Ansatz regt auch die Unterstützung der Konfigurationsmechanismen auf Metaebene an, wodurch die Konfiguration auch für nicht vordefinierte Sprachen anwendbar wird.

Allerdings finden sich Erweiterungen von Meta-CASE-Tools die Modelltransformationen ermöglichen (vgl. *Generic Modeling Environment* [ALSS03; LMBK01] und *Metaview* [ThMS03]). Dabei werden unter Rückgriff auf die Graphentheorie über eine algorithmische Suche strukturelle Muster im Modell der

Ausgangsmodellierungssprache identifiziert und in äquivalente Muster eines Modells der Zielmodellierungssprache transformiert.

Die Modelltransformation zielt auf die Überführung eines Ursprungsmodells in ein Zielmodell ab, wobei die Sprachen beider Modelle stark voneinander abweichen können. Dieses geschieht unter Rückgriff auf genau für diese Kombination der Ursprungs- und Zielsprache definierten Transformationsregeln [EHTE97]. Bspw. können Transformationen zwischen Ereignisgesteuerten Prozessketten und Petri-Netzen vorgenommen werden (vgl. z. B. [MoRo00]). Die Ansätze zur Modelltransformation zeichnen sich durch eine hohe Allgemeinheit der Operatoren zur Definition der Transformationsregeln aus (z. B. Create new, Replace, Same, Create Reference, Create Link, Delete, Refer else Create, Create inside, Refer to). Der Mächtigkeit und Flexibilität der Formulierung von Transformationen stehen hohe Kompetenzanforderungen an die Anwendung der Operatoren (insbesondere unter Beachtung der korrekten Reihenfolgen) gegenüber. Der hier vorgestellte Ansatz formuliert stattdessen speziellere Konfigurationsmechanismen, die aufgrund ihres höheren Domänenbezugs für den Anwender verständlicher sind. Im Sinne des Domain-Specific Modeling [NSKL98] sind die vorgestellten Konfigurationsmechanismen daher Spracherweiterungen, die speziell an die Anforderungen der konfigurativen Referenzmodellierung angepasst sind.

Der hier vorgestellte Ansatz basiert auf dem Konzept der Modellprojektion. Die Modellprojektion ermöglicht die Transformation von Ursprungsmodellen in Zielmodelle durch die Bildung von Sichten auf dem Ursprungsmodell. Dieses setzt jedoch voraus, dass die Modellierungssprache des Zielmodells eine Untermenge der Modellierungssprache des Ursprungsmodells ist. Die Modellprojektion lässt sich relativ leicht umsetzen, indem die Modelle als relationale Datenstrukturen abgebildet werden. In diesem Fall kann die Sichtenbildung über das Konzept der Datenbanksichten (Views) realisiert und daher nahezu vollständig durch Datenbankanweisungen implementiert werden.

Bei dem Ansatz dieses Beitrags wird der Suchraum im Vergleich zur Modelltransformation besser strukturiert, weil die betroffenen Modellelemente über die Konfigurationsregeln direkt identifizierbar sind. Dieses problemspezifische Wissen kann für eine effiziente Suche verwendet werden (vgl. Konsequenzen aus dem No-Free-Lunch-Theorem [WoMa97]). Die auf Modelltransformationen beruhenden Ansätze begegnen den ihnen innewohnenden Suchkomplexitätsproblemen zum Teil ebenfalls dadurch, dass sie die Verknüpfung von Mustern und zu transformierenden Modellteilen zulassen. Über diese „initial bindings“ [ALSS03] kommt es zu einer Annäherung der alternativen Ansätze.

5 Fazit

Konfigurative Referenzmodellierung kann die Effizienz der fachkonzeptionellen Modellierung erhöhen. Dabei werden die Referenzmodelle über Konfigurationsregeln an Anwendungskontexte angepasst und dann als Ausgangslösungen für die Entwicklung projektspezifischer Modelle bereitgestellt. Für die Beschreibung von Anwendungskontexten eignen sich Unternehmensmerkmalsausprägungen und Perspektiven. Die Modellierung konfigurierbarer Referenzmodelle erfordert die Erweiterung von Modellierungssprachen, die es ermöglichen, Anwendungskontexten verschiedene Referenzmodellvarianten zuzuordnen. Die Effizienz der Modellierung kann dabei durch Vorgabe von Konfigurationsmechanismen unterschiedlichen Wirkungsgrads gefördert werden. Die vorgeschlagenen Mechanismen generieren aus einem Gesamtmodell parameterspezifische Sichten.

Empfehlungen zu Modellierungstechniken für unterschiedliche Anwendungskontexte liegen zurzeit vorwiegend in Form von methodischen Informationssystemarchitekturen vor (Beispiele: [Sche01; Fran94; Zach87; Espr89]). Hierbei erfolgt die Differenzierung der Anwendungskontexte, wie z. B. Fachanwender, DV-Experte, Datenmanagement, Prozessmanagement, hauptsächlich über die Modelltypselektion. Die hier identifizierten und spezifizierten Konfigurationsmechanismen stellen einen geeigneten Rahmen bereit, um Modellierungs-Know-how in detaillierterer Form zu dokumentieren. Spezifische Modellierungstechniken für konkrete Konfigurationsparameterausprägungen lassen sich auf diese Weise übersichtlich beschreiben. (Vgl. Abbildung 9, die Konfigurationsparameterausprägungen – hier exemplarisch die Perspektiven Anwendungssystem- und Organisationsgestaltung und ihnen zugeordnete Subperspektiven – den Konfigurationsmechanismen gegenüberstellt und die Auswirkungen der Konfiguration auf die Modelltypen und Modelle dokumentiert.)⁵

⁵ Weitere praktische Anwendungsbeispiele finden sich ausführlich in [BeDK02] und in dem Beitrag von Becker und Knackstedt in diesem Band.

Konfigurationsmechanismus	Perspektive	Anwendungssystemgestaltung			Organisationsgestaltung		
		eEPK	Entity-Relationship-Modell	Anwendungssystemarchitekturmodell	eEPK	Fachbegriffsmodell	Organigramm
Modelltypselektion (zugelassene Modelltypen)							
Elementtypselektion (zugelassene Elementtypen je Modelltyp)		Funktion, Ereignis, Operator, Typ, Dartcluster, Anwendungssystemtyp	komplett	SE ohne Kompetenz DS komplett	Funktion, Ereignis, Operator, Fachbegriff(scluster), Organisationsobjekt	komplett	komplett
Elementselektion über Typen (zugelassene Beziehungstypen je Modelltyp)						PKR nur OObj-Zugehörigkeiten	WM ohne disziplinäre Weisungen PM komplett

Legende:

DS: Datenschutz

PKR: Prozesskostenrechnung

SE: Softwareentwicklung

OObj.: Organisationsobjekt

PM: Prozessmanagement

WM: Wissensmanagement

Abbildung 9: Überblick über eine konfigurative Referenzmodellierungstechnik

Literatur

- [Allw98] Allweyer, T.: Modellbasiertes Wissensmanagement. Information Management & Consulting, 13 (1998) 1, S. 37-45.
- [ALSS03] Agrawal, A.; Levendovszky, T.; Sprinkle, J.; Shi, F.; Karsai, G.: Generative Programming via Graph Transformations in the Model-Driven Architecture. http://www.isis.vanderbilt.edu/publications/archive/Agrawal_A_11_5_2002_Generative.pdf. Abrufdatum: 29.01.2003
- [BDKK02] Becker, J.; Delfmann, P.; Knackstedt, R.; Kuroпка, D.: Konfigurative Referenzmodellierung. In: Becker, J.; Knackstedt, R. (Hrsg.): Wissensmanagement mit Referenzmodellen. Konzepte für die Anwendungssystem- und Organisationsgestaltung. Heidelberg 2002, S. 25-144.
- [BeDK02] Becker, J.; Delfmann, P.; Knackstedt, R.: Eine Modellierungstechnik für die konfigurative Referenzmodellierung. In: Becker, J.; Knackstedt, R. (2002): Referenzmodellierung 2002. Methoden – Modelle – Erfahrungen. Arbeitsbericht des Instituts für Wirtschaftsinformatik, Nr. 90. Münster 2002, S. 35-79.
- [Blis00] Bliss, C.: Management von Komplexität. Ein integrierter, systemtheoretischer Ansatz zur Komplexitätsreduktion. Wiesbaden 2000.
- [BoJR98] Booch, G.; Jacobson, I.; Rumbough, J.: The Unified Modeling Language User Guide. Boston 1998.
- [Broc02] vom Brocke, J.: Referenzmodellierung. Gestaltung und Verteilung von Konstruktionsprozessen. Dissertation, Universität Münster 2002.
- [Chen76] Chen, P. P.: The Entity-Relationship Model. Toward a Unified View of Data. ACM Transactions on Database-Systems, 1 (1976) 1, S. 9-36.

- [DaDa97] Date, C.; Darwen, H.: A Guide to the SQL Standard. 4. Aufl., Boston 1997.
- [DaSh96] Darke, P.; Shanks, G.: Stakeholder Viewpoints in Requirements Definition. In: Requirements Engineering, 1 (1996) 1, S. 88-105.
- [EHTE97] Engels, G.; Heckel, R.; Taentzer, G.; Ehrig, H.: A View-Oriented Approach to System Modelling Based on Graph Transformation. ACM SIGSOFT Software Engineering Notes, 22 (1997) 6, S. 327-343.
- [Espr89] ESPRIT Consortium AMICE: Open System Architecture for CIM. Berlin u. a. 1989.
- [FeSi01] Ferstl, O. K.; Sinz, E. J.: Grundlagen der Wirtschaftsinformatik. Bd. 1, 4. Aufl., München, Wien 2001.
- [Fran94] Frank, U.: Multiperspektivische Unternehmensmodellierung. Theoretischer Hintergrund und Entwurf einer objektorientierten Entwicklungsumgebung. München, Wien 1994.
- [GeME03] Generischer Modelleditor. <http://wiseweb.wiwi.tu-dresden.de/gme/>. Abrufdatum: 29.01.2003.
- [Glas00] Von Glasersfeld, E.: Konstruktion der Wirklichkeit und des Begriffs der Objektivität. In: O. Hrsg.: Einführung in den Konstruktivismus. 5. Aufl., München, Zürich 2000, S. 9-39.
- [GrEs00] Greiffenberg, S.; Esswein, W.: Stand der Entwicklung einer Methode zur Metamodellierung. Arbeitsbericht des Lehrstuhls Wirtschaftsinformatik, insb. Systementwicklung, Technische Universität Dresden. Dresden 2000.
- [Groc82] Grochla, E.: Grundlagen der organisatorischen Gestaltung. Stuttgart 1982.
- [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: J. Akoka, M. Bouzeghoub, I. Comyn-Wattiau, E. Métais (Hrsg.): Conceptual Modeling – ER '99 – 18th International Conference on Conceptual Modeling, Paris, France, November 15-18, 1999 Proceedings. Berlin u. a. 1999, S. 203-217.
- [Holt01] Holten, R.: Metamodell. In: P. Mertens (Haupt Hrsg.): Lexikon der Wirtschaftsinformatik. 4. Aufl., Berlin u. a. 2001, S. 300-301.
- [HoVe96] Hofstede, A. H. M.; Verhoef, T. F.: Meta-CASE: Is the game worth the candle? Info Systems Journal, (1996) 6, S. 41-68.
- [JKSK00] Junginger, S.; Kühn, H.; Strobl, R.; Karagiannis, D.: Ein Geschäftsprozessmanagement-Werkzeug der nächsten Generation – ADONIS: Konzeption und Anwendungen. Wirtschaftsinformatik, 42 (2000) 5, S. 392-401.
- [KeNS92] Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozessmodellierung auf der Grundlage „Ereignisgesteuerter Prozessketten (EPK)“. Veröffentlichungen des Institutes für Wirtschaftsinformatik. Heft 89. Hrsg.: A.-W. Scheer. Saarbrücken 1992.

- [Kuge00] Kugeler, M.: Informationsmodellbasierte Organisationsgestaltung. Modellierungskonventionen und Referenzvorgehensmodell zur prozessorientierten Reorganisation. Berlin 2000.
- [Lang97] Lang, K.: Gestaltung von Geschäftsprozessen mit Referenzprozessbausteinen. Wiesbaden 1997.
- [LMBK01] Ledeczi, A.; Maroti, M.; Bakay, A.; Karsai, G.; Garrett, J.; Thomason, C.; Nordstrom, G.; Sprinkle, J.; Volgyesi, P.: The Generic Modeling Environment. In: Proceedings of WISP'2001. Budapest 2001.
- [LoHM02] Lohmann, M.; Hau, M.; Mertens, P.: Anforderungsanalyse auf der Basis von Unternehmensmerkmalen. In: J. Becker, R. Knackstedt (Hrsg.): Wissensmanagement mit Referenzmodellen. Heidelberg 2002, S. 279-289.
- [Loud94] Louden, K. C.: Programmiersprachen: Grundlagen – Konzepte – Entwurf. Bonn u. a. 1994.
- [Luhm99] Luhmann, N.: Zweckbegriff und Systemrationalität. Über die Funktion von Zwecken in sozialen Systemen. 6. Aufl., Frankfurt am Main 1999.
- [MBGH96] Mertens, P.; Bissantz, N.; Geyer, H.; Hagedorn, J.; Holzner, J.; Ludwig, M.: IV-Anwendungsarchitekturen für Branchen und Betriebstypen – erörtert am Beispiel der Ergebnisrechnung. In: Wirtschaftsinformatik, 38 (1996) 5, S. 485-495.
- [Meis01] Meise, V.: Ordnungsrahmen zur prozessorientierten Organisationsgestaltung. Modelle für das Management komplexer Reorganisationsprojekte. Hamburg 2001.
- [MeLo00] Mertens, P.; Lohmann, M.: Branche oder Betriebstyp als Klassifikationskriterien für die Standardsoftware der Zukunft? Erste Überlegungen, wie künftig betriebswirtschaftliche Standardsoftware entstehen könnte. In: F. Bodendorf, M. Grauer (Hrsg.): Verbundtagung Wirtschaftsinformatik 2000. Aachen 2000, S. 110-135.
- [MoRo00] Moldt, D.; Rodenhagen, J.: Ereignisgesteuerte Prozessketten und Petrinetze zur Modellierung von Workflows. In: H. Giese, S. Philippi (Hrsg.): Visuelle Verhaltensmodellierung verteilter und nebenläufiger Software-Systeme. 8. Workshop des Arbeitskreises GROOM der GI Fachgruppe 2.1.9 Objektorientierte Software-Entwicklung, 13.-14. November 2000. Münster 2000, S. 57-63.
- [NSKL98] Nordstrom, G.; Sztipanovits, J.; Karsai, G.; Ledeczi, A.: Metamodeling – Rapid Design and Evolution of Domain-Specific Modeling Environments, Proc. IEEE ECBS'98 Conf., 1998.
- [Ortn97] Ortner, E.: Methodenneutraler Fachentwurf. Stuttgart, Leipzig 1997.
- [Remm97] Remme, M.: Konstruktion von Geschäftsprozessen. Ein modellgestützter Ansatz durch Montage generischer Prozesspartikel. Wiesbaden 1997.
- [Rohl95] Rohloff, M.: Produktionsmanagement in modularen Organisationsstrukturen. Reorganisation der Produktion und objektorientierte Informationssysteme für verteilte Planungssegmente. München, Wien 1995.
- [Rose96a] Rosemann, M.: Komplexitätsmanagement in Prozessmodellen. Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung. Wiesbaden 1996.

- [Rose96b] Rosemann, M.: Multiperspektivische Informationsmodellierung auf Basis der Grundsätze ordnungsmäßiger Modellierung. In: Management & Computer, 4 (1996) 4, S. 219-226.
- [Rose98a] Rosemann, M.: Managing the Complexity of Multiperspective Information Models using the Guidelines of Modeling. In: D. Fowler, L. Dawson (Hrsg.): Proceedings of the 3rd Australian Conference on Requirements Engineering. Geelong 1998, S. 101-118.
- [Rose98b] Rosemann, M.: Die Grundsätze ordnungsmäßiger Modellierung. Intention, Entwicklung, Architektur und Multiperspektivität. In: Maicher, M.; Scheruhn, H.-J. (Hrsg.): Informationsmodellierung. Referenzmodelle und Werkzeuge. Wiesbaden 1998, S. 1-21.
- [Sche01] Scheer, A.-W.: ARIS – Modellierungsmethoden, Metamodelle, Anwendungen. 4. Auflage, Berlin u. a. 2001.
- [Sche96] Scheer, A.-W.: ARIS-Toolset: Vom Forschungs-Prototypen zum Produkt. Informatik-Spektrum, 19 (1996) 2, S. 71-78.
- [Schl00] Schlagheck, B.: Objektorientierte Referenzmodelle für das Prozess- und Projektcontrolling. Grundlagen – Konstruktion – Anwendungsmöglichkeiten. Wiesbaden 2000.
- [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. Aufl., Stuttgart 1983.
- [Stra96] Strahringer, S.: Metamodellierung als Instrument des Methodenvergleichs. Eine Evaluierung am Beispiel objektorientierter Analysemethoden. Aachen 1996.
- [ThMS03] The Metaview System. <http://web.cs.ualberta.ca/~softeng/Metaview/system/system.shtml>. Abrufdatum: 29.01.2003.
- [Wolf01] Wolf, S.: Wissenschaftstheoretische und fachmethodische Grundlagen der Konstruktion von generischen Referenzmodellen betrieblicher Systeme. Aachen 2001.
- [Woll86] Wollnik, M.: Implementierung computergestützter Informationssysteme. Berlin, New York 1986.
- [WoMa97] Wolpert, D.; Macready, W.: No Free Lunch Theorems for optimization. IEEE Transactions on Evolutionary Computation, 1 (1997) 1, S. 67-82.
- [Zach87] Zachman, J. A.: A Framework for Information Systems Architecture. IBM Systems Journal, 26 (1987) 3, S. 277-293.