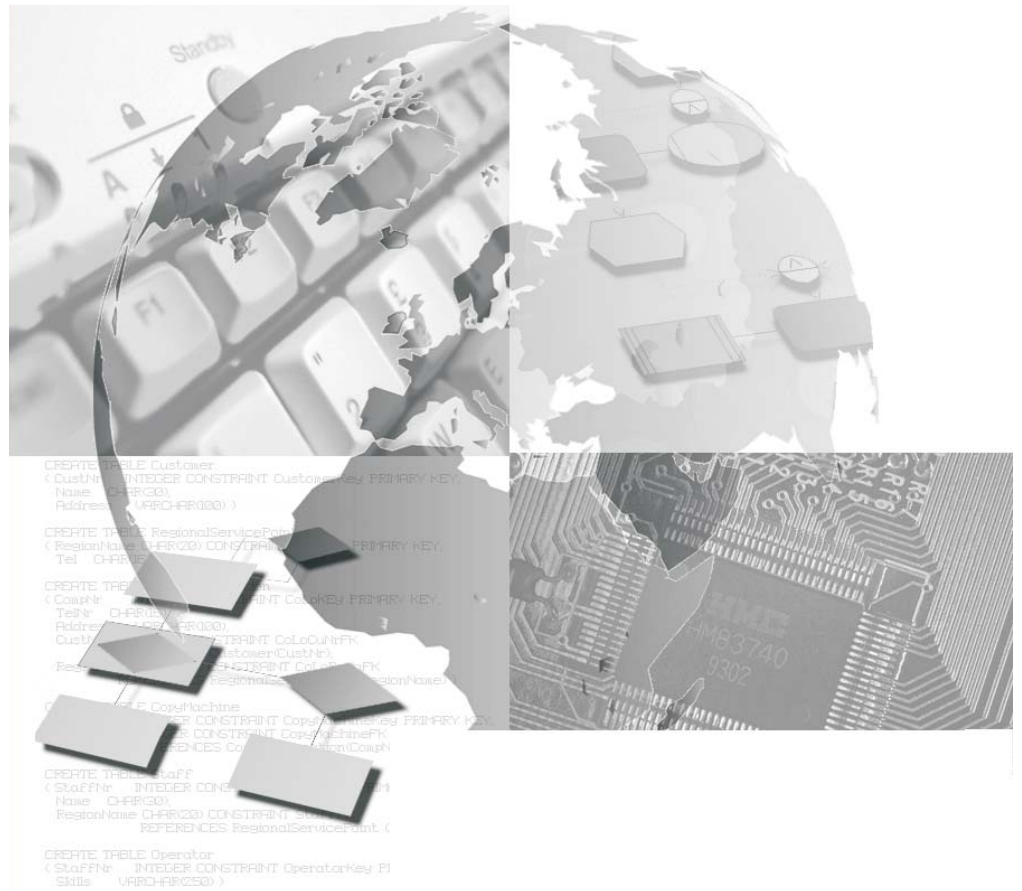




Arbeitsberichte



Arbeitsbericht Nr. 116

Modellierung und Konfiguration elektronischer Geschäftsdokumente mit dem H2-Toolset

Jörg Becker, Stephan Kramer, Christian Janiesch

Arbeitsberichte des Instituts für Wirtschaftsinformatik
Herausgeber: Prof. Dr. J. Becker, Prof. Dr. H. L. Grob, Prof. Dr. S. Klein,
Prof. Dr. H. Kuchen, Prof. Dr. U. Müller-Funk, Prof. Dr. G. Vossen

Arbeitsbericht Nr. 116

**Modellierung und Konfiguration
elektronischer Geschäftsdokumente mit
dem H2-Toolset**

Jörg Becker, Stephan Kramer, Christian Janiesch

ISSN 1438-3985

Inhaltsverzeichnis

Inhaltsverzeichnis	II
Abbildungsverzeichnis	IV
Tabellenverzeichnis	VI
Abkürzungsverzeichnis	VII
1 Potenziale und Herausforderungen bei der Nutzung elektronischer Geschäftsdokumente	1
2 Evolutionäre Entwicklung von EDI-Austauschstandards	4
2.1 Grundlagen	4
2.2 Traditionelle Ansätze	5
2.2.1 Ausgangssituation	5
2.2.2 UN/EDIFACT	7
2.3 Aktuelle Ansätze	8
2.3.1 Ausgangssituation	8
2.3.2 RosettaNet	9
2.4 Ansätze der nächsten Generation	11
2.4.1 Ausgangssituation	11
2.4.2 Core Components Technical Specification	12
2.4.3 Universal Business Language	22
2.4.4 Gegenüberstellung und Bewertung von CCTS und UBL	29
3 Konfiguration elektronischer Geschäftsdokumente mit dem H2-Toolset	33
3.1 Referenzmodelladaption	33
3.1.1 Generische Referenzmodelladaptionstechniken	34
3.1.2 Konfigurative Referenzmodelladaptionstechniken	35
3.1.3 Konfigurative Referenzmodellierung mit dem H2-Toolset	38
3.2 Sprachspezifikation im H2-Toolset	41
3.2.1 Modellierung der CCTS	41
3.2.2 Modellierung der UBL	51
3.3 Exemplarische Umsetzung zweier Geschäftsdokumente	56
3.3.1 Modellierung der ACC Address in CCTS	56
3.3.2 Modellierung des Geschäftsdokuments Order Response Simple in UBL	61
3.4 Einordnung der Kontextkategorien in die Konfigurationsparameterstruktur der konfigurativen Referenzmodellierung	63
3.5 Modellkonfiguration	66
3.5.1 Evaluation der Möglichkeiten anhand einer Fallstudie	66
3.5.2 Evaluation der Grenzen	80
3.5.3 Evaluation der Potenziale	85
4 Fazit und Ausblick	88
Literaturverzeichnis	90
Anhang	95
A ebXML-Framework	95
B Übernahme potenzieller ABIE in das Repository	98
C Fachkonzeptionelle Spezifikation konfigurativer Referenzmodellierungsmechanismen	100

Abbildungsverzeichnis

Abb. 1:	Interoperabilität traditioneller Ansätze auf syntaktischer und semantischer Ebene	8
Abb. 2:	Interoperabilität aktueller Ansätze auf syntaktischer und semantischer Ebene	11
Abb. 3:	Kontextspezifische Adaption von semantischen Blöcken	13
Abb. 4:	ACC Financial Account mit zugehörigen BCC und ASCC	15
Abb. 5:	Schematische Darstellung der Aggregationsbeziehungen zwischen Core Components	16
Abb. 6:	ABIE Credit_Financial Account	17
Abb. 7:	Business Data Type mit annotierten Restriktionen	18
Abb. 8:	Schematische Darstellung der Aggregationsbeziehungen zwischen Business Information Entities	19
Abb. 9:	Übergeordneter Zusammenhang der CCTS-Komponenten	22
Abb. 10:	UBL-Geschäftsdokument Order Response Simple	24
Abb. 11:	Bestellprozess in UBL	24
Abb. 12:	UBL-Aggregate Business Information Entity Financial Account	26
Abb. 13:	Abgeleitetes UBL-Aggregate Business Information Entity Auction_Financial Account	26
Abb. 14:	Erweitertes UBL-Aggregate Business Information Entity Internal_Financial Account	27
Abb. 15:	Modellierter Typ ProductDescriptionType	28
Abb. 16:	Dokumentation von Kontextkategorien und -werten	29
Abb. 17:	Interoperabilität von Ansätzen der nächsten Generation auf syntaktischer und semantischer Ebene	30
Abb. 18:	Kontextfreie Grammatik zur Formulierung von Konfigurationstermen	37
Abb. 19:	Hauptkomponenten des H2-Kernrepositories	39
Abb. 20:	Sprach- und Modelleditor im H2-Toolset	41
Abb. 21:	ERM des CCTS-Konzepts	42
Abb. 22:	Identifikation des impliziten Representation Terms	43
Abb. 23:	ERM des adaptierten CCTS-Konzepts für das H2-Toolset	44
Abb. 24:	Visualisierender Aspekt der Kardinalitäten im Modelleditor	46
Abb. 25:	Kontext- und Elementstruktur der CCTS	47
Abb. 26:	Endlosschleife durch gegenseitige Referenzierung	49
Abb. 27:	Objekttypen und Kontexte der CCTS im Spracheditor	50
Abb. 28:	ERM des UBL-Konzepts	51
Abb. 29:	ERM des adaptierten UBL-Konzepts für das H2-Toolset	52
Abb. 30:	Kontext- und Elementstruktur der UBL	52

Abb. 31:	Modellierte Attribute für Objekttypen der UBL.....	54
Abb. 32:	Objekttypen und Kontexte der UBL im Spracheditor	55
Abb. 33:	Modellierung von BCC	57
Abb. 34:	Annotation einer ACC-Occurrence unter eine ASCC	57
Abb. 35:	Modellierung von Primitive Types und Core Data Types.....	58
Abb. 36:	ACC Address der Core Component Library im H2-Toolset	60
Abb. 37:	Business Document Order Response Simple im H2-Toolset.....	62
Abb. 38:	Modell der Konfigurationsparameter von CCTS bzw. UBL im H2-Toolset	65
Abb. 39:	Geschäftsmodell der Panther GmbH	67
Abb. 40:	Konfigurationsparameter der Geschäftssituation im H2-Toolset.....	68
Abb. 41:	Konfigurationsterme für einzelne Geschäftsszenarien	69
Abb. 42:	Ontologische Beziehungen der Geschäftsszenarien	70
Abb. 43:	Zuweisung des Terms zu irrelevanten Elementen aller Geschäftsszenarien	71
Abb. 44:	Zuweisung von Termen für Handel mit deutschen Großhändlern über ein webbasiertes Informationssystem.....	73
Abb. 45:	Zuweisung von Termen für Handel mit französischen Zwischenhändlern über einen Internetshop	75
Abb. 46:	Kombination der Konfigurationsterme.....	76
Abb. 47:	Zuweisung von Termen für direkten Handel mit amerikanischen Zwischenhändlern.....	77
Abb. 48:	Darstellungsvariation beim Übergang von Core Components zu Business Information Entities	78
Abb. 49:	Referenzen der Supplier und Customer Party	81
Abb. 50:	Abdeckung des CCTS-Konzepts durch das H2-Toolset	87
Abb. 51:	Durch die ebXML-Spezifikationen unterstützte Aktivitäten.....	96
Abb. 52:	Vergleich potenzieller ABIE mit existierenden ABIE innerhalb des Repositories in Form einer EPK.....	99
Abb. 53:	Meta-Metamodellkonstrukte zur Modelltypselektion	100
Abb. 54:	Meta-Metamodellkonstrukte zur Elementtypselektion.....	101
Abb. 55:	Meta-Metamodellkonstrukte zur Elementselektion nach Termen.....	102
Abb. 56:	Metamodellkonstrukte zur Konfiguration nach Termen	103
Abb. 57:	Meta-Metamodellkonstrukte zur Bezeichnungsvariation.....	103
Abb. 58:	Meta-Metamodellkonstrukte zur Darstellungsvariation.....	104

Tabellenverzeichnis

Tab. 1:	Semiotische Ebenen von EDI	4
Tab. 2:	Überblick der Ansätze	5
Tab. 3:	Dimensionen und Beispiele traditioneller EDI-Austauschstandards.....	7
Tab. 4:	Kontextkategorien der CCTS	20
Tab. 5:	Übersicht über die in <i>Address</i> enthaltenen Elemente und Häufigkeiten ..	59
Tab. 6:	Übersicht über die in <i>Order Response Simple</i> enthaltenen Elemente und Häufigkeiten	61
Tab. 7:	Klassifikationsschema der Kontextkategorien	64
Tab. 8:	Kontextkategorien der Geschäftssituation.....	68
Tab. 9:	Irrelevante Elemente für alle Geschäftsszenarien.....	71
Tab. 10:	Irrelevante Elemente für Handel mit deutschen Großhändlern über ein webbasiertes Informationssystem.....	72
Tab. 11:	Irrelevante Elemente für Handel mit französischen Zwischenhändlern über einen Internetshop	74
Tab. 12:	Irrelevante Elemente für direkten Handel mit amerikanischen Zwischenhändlern.....	76
Tab. 13:	Reduktion von <i>Order Response Simple</i> in den einzelnen Konfigurationskontexten	79
Tab. 14:	Eigenschaften der Ansätze im Überblick	88

Abkürzungsverzeichnis

ABIE	Aggregate Business Information Entity
ACC	Aggregate Core Component
ANSI	American National Standards Institute
ASBIE	Association Business Information Entity
ASCC	Association Core Component
ASCII	American Standard Code for Information Interchange
B2B	Business to Business
BBIE	Basic Business Information Entity
BCC	Basic Core Component
BDT	Business Data Type
BIE	Business Information Entity
CC	Core Component
CCL	Core Component Library
CCTS	Core Components Technical Specification
CDT	Core Data Type
DEN	Dictionary Entry Name
DT	Data Type
ebXML	Electronic Business XML
EDI	Electronic Data Interchange
EPK	Ereignisgesteuerte Prozesskette
ERM	Entity-Relationship-Modell
ET	Entitytyp
HTML	Hypertext Markup Language
KMU	Kleine und mittlere Unternehmen
OAGIS	Open Application Group Integration Specification
OASIS	Organization for the Advancement of Structured Information Standards
OC	Object Class
PIP	Partner Interface Processes
PT	Property Term
QT	Qualifier Term
RT	Representation Term
SGML	Standard Generalized Markup Language
SWIFT	Society for Worldwide Interbank Financial Telecommunication
TCP/IP	Transmission Control Protocol / Internet Protocol
TRADACOMS	Trade Data Communication Standards
UBL	Universal Business Language
UML	Unified Modeling Language
UMM	UN/CEFACT Modeling Methodology
UN/CEFACT	United Nations Centre for Trade Facilitation and Electronic Business
UNSM	United Nations Standard Message
VDA	Verband der Automobilindustrie
WWW	World Wide Web
XML	eXtensible Markup Language
XSD	XML Schema Definition

1 Potenziale und Herausforderungen bei der Nutzung elektronischer Geschäftsdokumente

Die Entwicklung und Einführung des *elektronischen Datenaustauschs (Electronic Data Interchange, EDI)* Ende der sechziger Jahre des letzten Jahrhunderts haben zu einer Verlagerung der Geschäftsbeziehungen und -dokumente von traditionellen auf elektronische Medien geführt. Diese Entwicklung beinhaltet enormes wirtschaftliches Potenzial. EDI erhöht die Geschwindigkeit und Effizienz des Austauschs und ermöglicht darüber hinaus die Vereinfachung und Restrukturierung innerbetrieblicher Abläufe.¹ Traditionelle EDI-Ansätze aus früherer Zeit gelten jedoch als proprietär, inflexibel und aufwändig zu implementieren. Beim Datenaustausch müssen die übertragenen Geschäftsdokumente auf die Datenstrukturen der internen Systeme abgebildet werden. Dieser Prozess verursacht einen hohen Zeit- und Kostenaufwand und beinhaltet ein erhöhtes Fehlerpotenzial. Die mangelnde Interoperabilität der Systeme erfordert daher eine hohe Frequenz von Transaktionen und zeitliche Bindung der Geschäftspartner, um die angefallenen Kosten der Investition in EDI-Systeme und des *Mappings* zu amortisieren.

Die generellen Vorteile automatisiert abgewickelter elektronischer Geschäftsprozesse in Kombination mit den Nachteilen traditioneller Austauschstandards haben zu einer evolutionären Entwicklung von EDI-Standards geführt. Ziel dieser Bemühungen war und ist es, die Interoperabilität der Standards zu verbessern und so den Implementierungs- und Mappingaufwand zu reduzieren. Mit der Entwicklung und Verbreitung der *eXtensible Markup Language (XML)* entstand die Hoffnung, das Problem der Dateninteroperabilität durch ihre Etablierung als „Lingua Franca“ des Internets zu lösen.² XML ist jedoch lediglich eine Sprache, und kann von den drei semiotischen Ebenen³ Syntax, Semantik und Pragmatik nur die Syntax vereinheitlichen. Eine Spezifikation der semantischen Ebene elektronischen Datenaustauschs findet nicht statt, die Modellierung der Struktur unterliegender Geschäftsdaten bleibt demnach unberücksichtigt. Dies führt in der Praxis dazu, dass Unternehmen, die ihren Datenaustausch auf Basis von XML vornehmen, weiterhin ein Mapping der ausgetauschten Geschäftsdaten vorzunehmen haben, da unterschiedliche XML-Dialekte und -vokabulare sowohl unternehmensindividuell innerhalb einer Branche als auch branchenübergreifend verwendet werden. Deren Feldinhalte innerhalb der Geschäftsdokumente sind jedoch nicht unmittelbar miteinander kompatibel. *Das ursprüngliche Problem aus den Anfangszeiten von EDI konnte somit nicht gelöst werden, sondern verschob sich nur auf eine andere Ebene.*

¹ Vgl. Emmelhainz (1990), S. 20.

² Vgl. Weitzel; Harder; Buxmann (2001), S. 10.

³ Zu den semiotischen Ebenen vgl. Mittelstraß (1995), S. 781 ff.

Um die Interoperabilität von elektronischen Geschäftsdokumenten zu gewährleisten, muss zwischen den austauschenden Akteuren *Einigkeit über die syntaktische und die semantische Struktur bestehen*. Durch die starke Verbreitung von XML herrscht auf syntaktischer Ebene mittlerweile weitgehende Übereinstimmung. Auf semantischer Ebene existiert bisher keine Einigkeit. Daher ist es erforderlich, diese durch die Spezifikation eines Standards sicherzustellen, um eine unkontrollierte Ausbreitung individueller semantischer Strukturen zwischen den Marktteilnehmern zu verhindern. Eine Initiative des *United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT)* und der *Organization for the Advancement of Structured Information Standards (OASIS)* entwickelt unter dem Namen *ebXML (Electronic Business XML)* eine breite Palette verschiedener Standards, um den elektronischen Handel von Unternehmen jeder Größe und jeder geographischen Region zu unterstützen.⁴ Einer dieser Standards ist die *Core Components Technical Specification (CCTS)*. Die CCTS ist implementierungsunabhängig und beinhaltet eine Methodologie für die Entwicklung kontextspezifischer semantischer Einheiten und Blöcke aus kontextneutralen Vorlagen, die die Erstellung neuer sowie die Restrukturierung bestehender Geschäftsvokabularien ermöglichen.⁵ Die branchenspezifische Anpassung der Daten erfolgt durch ein kontextgetriebenes Prinzip zur Variantengenerierung. Die Anwendung der CCTS ermöglicht den teilnehmenden Unternehmen, die semantische Äquivalenz ausgetauschter Geschäftsdokumente zu den eigenen Datenstrukturen *automatisiert* festzustellen und auf diese zu mappen. Die syntaktische und semantische Interoperabilität der Geschäftsdokumente ist somit gewährleistet. Eine Sprache, die die CCTS zu großen Teilen implementiert, ist die *Universal Business Language (UBL)*. Die Variantengenerierung in UBL bzw. CCTS ist jedoch noch nicht vollständig ausgereift und beinhaltet Fehlerpotenziale, die aufwändige und inkonsistente Modellierung sowie redundante Speicherung der Varianten fördern.

Einen potenziellen Lösungsansatz für diese Probleme bietet die *konfigurative Referenzmodellierung*. Dabei werden kontextspezifische Varianten durch eine regelbasierte Projektion eines für den Anwendungsbereich vollständigen Ausgangsmodells generiert. Ziel dieses Arbeitsberichts ist es, die *Übertragbarkeit des Ansatzes der konfigurativen Referenzmodellierung auf den Bereich der Variantengenerierung von elektronischen Geschäftsdokumenten mittels des softwarebasierten Metamodellierungswerkzeugs H2-Toolset zu untersuchen*. Dabei soll vorrangig geprüft werden, ob eine Integration der Ansätze der konfigurativen Referenzmodellierung und der CCTS bzw. UBL einen Lösungsweg bietet, um die skizzierten Defizite der konventionellen Vorgehensweise nach CCTS bzw. UBL bei der Erstellung von elektronischen Geschäftsdokumentvarianten zu beseitigen. Es ist zu prüfen, inwieweit die regelbasierte Projektion eines allumfassenden Referenz-Geschäftsdokument zu einer

⁴ Vgl. ebXML (2006a).

⁵ Vgl. UN/CEFACT (2003a), S. 7.

kontextspezifischen Variante den Modellierungsaufwand reduzieren kann und welche weiteren Maßnahmen nötig sind, um die vorkonfigurierte Variante tatsächlich einzusetzen.

Die Ziele und zu untersuchenden Fragestellungen legen folgenden Aufbau des Arbeitsberichts nahe: In Kapitel 2 erfolgt ein Überblick über die evolutionäre Entwicklung von EDI-Austauschstandards. Zu diesem Zweck werden zunächst die relevanten Grundlagen der Standards skizziert (Kap. 2.1), um unterteilt in traditionelle Ansätze (Kap. 2.2), aktuelle Ansätze (Kap. 2.3) und Ansätze der nächsten Generation (Kap. 2.4) näher untersucht, und in die zentrale Fragestellung der Interoperabilität syntaktischer und semantischer Art eingeordnet zu werden. Da die syntaktischen und semantischen Eigenschaften der CCTS und der sie implementierenden UBL besonders viel versprechend erscheinen, stehen sie im Fokus des Kapitels. Diesen Ausführungen folgt der praktische Teil des Arbeitsberichts, die fachkonzeptionelle Spezifikation und exemplarische Umsetzung der Modellierung und Konfiguration elektronischer Geschäftsdokumente im H2-Toolset (Kap. 3). Kapitel 3.1 erläutert hierzu zunächst das Konzept der Referenzmodelladaption mit einem kurzen Überblick über die generischen und die konfigurativen Adaptionstechniken sowie deren Umsetzung im H2-Toolset. Im Folgenden wird eine Modellierungssprache für CCTS und UBL erstellt (Kap. 3.2) und für beide Ansätze jeweils ein elektronisches Geschäftsdokument modelliert (Kap. 3.3). CCTS und UBL spezifizieren unternehmensindividuelle Gegebenheiten durch Kontextkategorien und entsprechende Ausprägungen.⁶ Daher wird untersucht, in wieweit dieses Prinzip mit Unternehmensmerkmalen und Perspektiven der konfigurativen Referenzmodellierung übereinstimmt, welche Unterschiede existieren, und welche Probleme bei der Transformation entstehen (Kap. 3.4). Damit sind die Voraussetzungen für die Konfiguration der Referenz-Geschäftsdokumente erfüllt (Kap. 3.5). Im Fokus der Betrachtung steht dabei insbesondere die Fragestellung nach den Grenzen, den Möglichkeiten und den Potenzialen der konfigurativen Referenzmodellierung mit dem H2-Toolset. Kapitel 4 zieht ein Fazit der Ergebnisse dieses Berichts und unterzieht die verwendeten Methoden einer kritischen Würdigung. Ein Ausblick auf künftigen Forschungsbedarf bildet den Schluss.

⁶ Ältere Veröffentlichungen zu diesem Thema verwenden synonym die Bezeichnung *Context Driver*, vgl. bspw. ebXML (2001b), S. 6.

2 Evolutionäre Entwicklung von EDI-Austauschstandards

2.1 Grundlagen

Die Entwicklung einer einheitlichen Definition und Abgrenzung von EDI ist bis heute nicht zu erkennen.⁷ KLEIN definiert *elektronischen Datenaustausch* als „den zwischenbetrieblichen, elektronischen Austausch standardisierter Geschäftsdokumente zwischen Programmen“.⁸ Weitere Definitionen sind in ihrer grundlegenden Auffassung zwar ähnlich, grenzen die Dimensionen jedoch unterschiedlich ab.⁹ Im Kontext von EDI ist eine dokumentenorientierte Sichtweise von grundlegender Bedeutung.¹⁰ Dabei existieren für den Begriff des *Geschäftsdokuments* wiederum unterschiedliche Auffassungen. So definieren WEITZEL, HARDER und BUXMANN ein Geschäftsdokument als Abstraktion zur Beschreibung realer Austauschprozesse.¹¹ Ergänzend kann ein Geschäftsdokument als eine zweckgebundene und geschlossene Sammlung von Informationen definiert werden.¹² Aus semiotischer Sicht können für den elektronischen Datenaustausch die drei Teilbereiche *Syntax*, *Semantik* und *Pragmatik* unterschieden werden, wobei die Syntax durch die *Beziehungen zwischen Zeichen*, die Semantik durch die *Zeichenbedeutung* und die Pragmatik durch die *Handlungsmuster von Benutzern der Zeichen in einer bestimmten Situation* definiert werden.¹³ Die Bedeutung der semiotischen Ebenen für EDI ist Tab. 1 zu entnehmen.

Semiotische Ebene	Bedeutung für EDI
Pragmatik	Handlungs- und Reaktionsmuster, z. B. unternehmensindividuelle Ablaufszenarien
Semantik	Feldinhalte, z. B. mit ebXML / CCTS
Syntax	Nachrichtenstruktur, z. B. mit XML

Tab. 1: Semiotische Ebenen von EDI¹⁴

Die pragmatische Ebene beinhaltet vorwiegend interne Handlungs- und Reaktionsmuster der Unternehmen. Aufgrund der Fokussierung auf die technischen Aspekte von EDI wird sie bei den folgenden Untersuchungen nicht berücksichtigt.

⁷ Vgl. Lamprecht (1998), S. 15.

⁸ Klein (1998), S. 61.

⁹ Weitere EDI-Definitionen können bspw. Pfeiffer (1992), S. 18 f. oder Emmelhainz (1990), S. 4 f. entnommen werden.

¹⁰ Vgl. Weitzel; Harder; Buxmann (2001), S. 11.

¹¹ Weitzel; Harder; Buxmann (2001), S. 11.

¹² Vgl. Glushko; McGrath (2005), S. 11.

¹³ Vgl. Mittelstraß (1995), S. 784.

¹⁴ In Anlehnung an Klein (1998), S. 68.

Um die Interoperabilität von Geschäftsdokumenten zu gewährleisten, ist somit eine Übereinkunft über die verwendete Syntax und Semantik zu erzielen. Eine vereinheitlichte Syntax ermöglicht einen Austausch mit einem standardisierten Zeichensatz, während die Semantik die Modellierung der unterliegenden Geschäftsdokumente berücksichtigt und so ein Mapping der ausgetauschten Geschäftsdaten entfällt. Seit der Einführung von EDI Ende der sechziger Jahre des letzten Jahrhunderts durchlaufen dessen Austauschstandards eine evolutionäre Entwicklung.¹⁵ Den monolithischen, proprietären und inflexiblen Standards wie EDIFACT oder ANSI X 12 folgten Ansätze wie RosettaNet mit dem Ziel, die gestiegenen Anforderungen an Interoperabilität im globalisierten E-Business besser abzudecken. Obwohl diese die Anforderungen in vieler Hinsicht besser erfüllen als ihre Vorgänger, decken sie doch nicht das gesamte Spektrum wünschenswerter Eigenschaften ab. So besteht bei ihnen durch die Verwendung von XML zwar Einigkeit über die Syntax; die Modellierung der unterliegenden Geschäftsdaten als Vokabular eines Standards divergiert jedoch erheblich. Semantische Interoperabilität zwischen den Standards ist nicht gewährleistet. Die sich momentan in der Entwicklung befindenden Ansätze der „nächsten Generation“ wie CCTS und UBL sollen dieses Problem lösen. Um die fortlaufende Entwicklung der Standards zu verdeutlichen, folgt eine tabellarische Übersicht der einzelnen Ansätze (vgl. Tab. 2) und eine detaillierte Einordnung der EDI-Austauschstandards in ihren historischen Kontext sowie ein Vergleich ihrer Interoperabilitätsfähigkeiten auf syntaktischer und semantischer Ebene.

	Traditionell	Aktuell	Nächste Generation
Stellvertreter	VDA SWIFT EDIFACT ...	RosettaNet OAGIS 8.0 ...	UBL 2.0 OAGIS 9.0 ...
Interoperabilität	begrenzt	Syntax	Syntax + Semantik

Tab. 2: Überblick der Ansätze

2.2 Traditionelle Ansätze

2.2.1 Ausgangssituation

Die ersten EDI-Lösungen Ende der sechziger Jahre des letzten Jahrhunderts resultierten aus der Tatsache, dass ein Großteil der zwischenbetrieblichen Kommunikation noch schriftlich und über mehrere Stationen erfolgte. Dabei wurden die Dokumente in den betei-

¹⁵ Vgl. im Folgenden Janner et al. (2006), S. 1.

lichten Unternehmen erstellt, ausgedruckt und an die Kommunikationspartner geschickt, wo eine erneute Eingabe in die lokalen Systeme erfolgte.¹⁶ Die Fehleranfälligkeit und Ineffizienz dieser Mehrfacheingaben führten zur Entwicklung von Methoden, die diesen Austausch vollautomatisiert und ohne Medienbrüche ermöglichten. Die im Vergleich zum schriftlichen Austausch realisierten Vorteile sind vielschichtig: So sollen z. B. Erfassungsfehler und -kosten reduziert, interne Geschäftsprozesse optimiert, Lagerbestände reduziert, die Kapitalbindung gesenkt und auf veränderte Wettbewerbssituationen schneller reagiert werden können.¹⁷ In quantitativen Größen wird dabei von Kostenreduktionen im Verhältnis von zehn zu eins ausgegangen.¹⁸ Um die korrekte Übertragung zwischen Sender und Empfänger sicherzustellen, sind Absprachen über Übertragungsprotokolle und Datenformate der ausgetauschten Nachricht erforderlich. Theoretisch wären derartige Absprachen über individuelle Vereinbarungen zwischen den Kommunikationspartnern zu realisieren. Da aber eine Vielzahl begleitender und anpassender Maßnahmen durchgeführt, und diese in einem aufwändigen Prozess ausgehandelt werden müssen, fällt der monetäre Aufwand entsprechend hoch aus.¹⁹ Daher gilt es, individuelle Absprachen durch standardisierte Vereinbarungen zu ersetzen, damit pro Unternehmen nur eine einzelne Anpassung an diesen Standard vorzunehmen ist.²⁰ *Standards bilden somit die wirtschaftliche Grundlage für die Einführung von EDI.*²¹

In der Realität existieren viele unterschiedliche Nachrichtenstandards für den Austausch von Geschäftsdokumenten, die sich in die Dimensionen *Anwendungsbereich* und *Reichweite* unterteilen lassen (vgl. Tab. 3). So haben sich in einzelnen Branchen auf nationaler Ebene spezifische Standards wie *VDA (Verband der Automobilindustrie)* oder *TRADACOMS (Trade Data Communication Standards)* etabliert. Daneben existieren branchenspezifische, jedoch international gültige Standards wie *SWIFT (Society for Worldwide Interbank Financial Telecommunication)*. Branchenneutrale und national beschränkte Standards wie *ANSI X 12* vom *American National Standards Institute* sind ebenfalls etabliert. Diese Standards haben in ihren jeweiligen Domänen einen hohen Verbreitungsgrad erreicht. Sie eignen sich jedoch nicht für internationales und industrieübergreifendes E-Business, da ihre syntaktische und semantische Struktur zu unterschiedlich ist. Daher wurde ein branchenunabhängiger und internationaler Standard unter der Leitung der Vereinten Nationen entwickelt: *UN/EDIFACT (United Nations Rules for Electronic Data Interchange For Administration, Commerce and Transport)*.

¹⁶ Vgl. Weitzel; Harder; Buxmann (2001), S. 6.

¹⁷ Vgl. Emmelhainz (1990), S. 8 f., Weitzel; Harder; Buxmann (2001), S. 7, Thomas (1999), S. 7 sowie Sedran (1991), S. 18 f.

¹⁸ Vgl. Emmelhainz (1990), S. 7.

¹⁹ Vgl. Picot; Neuburger; Niggel (1991), S. 23.

²⁰ Vgl. Neuburger (1994), S. 19.

²¹ Picot; Neuburger; Niggel (1991), S. 23.

	Branchenspezifisch	Branchenneutral
National	VDA (Automobilbranche, D) TRADACOMS (Handel, GB)	ANSI X 12 (USA)
International	EDIFACT-Subsets, z. B. EANCOM SWIFT	UN/EDIFACT

Tab. 3: Dimensionen und Beispiele traditioneller EDI-Austauschstandards

2.2.2 UN/EDIFACT

EDIFACT hat, wie nahezu alle traditionellen Ansätze, eine eigene Nachrichtenstruktur und Semantik.²² Die Syntax ist nicht menschenlesbar und beinhaltet Segmente codiert in *ASCII* (*American Standard Code for Information Interchange*), Datenelementen, Datenelementgruppen, Bezeichnern, Werten und Metadaten, die durch eine Trennpunktsyntax strukturiert werden.²³ *Syntaktische Interoperabilität zu anderen Standards existiert aufgrund der eigenständigen Definition dieses spezifischen Austauschformats nicht.* Die Abstimmung von *Feldinhalten* kann mittels zweier Vorgehensweisen erfolgen: Zum einen können bereits existierende branchen- oder länderspezifische Austauschstandards in EDIFACT-Nachrichten konvertiert werden. Dabei sind logisch zusammengehörige Datenelemente zu Segmenten und Segmente wiederum zu Nachrichten zusammenzufassen.²⁴ Die Abstimmung der ausgetauschten Formate ist dabei zwischen den beteiligten Unternehmen individuell abzustimmen. Semantische Interoperabilität existiert bei diesen Individuallösungen nicht, da keine Schnittstellen zu anderen Standards existieren. Die zweite Möglichkeit besteht in der Verwendung vereinbarter Standardnachrichten, den *United Nations Standard Messages (UNSM)*. Diese sind in international abgestimmter Arbeit in Expertengremien entwickelt, verabschiedet und von den Vereinten Nationen zur Anwendung empfohlen worden.²⁵ Die Nachrichten enthalten eine Vielzahl von Segmenten und Datenelementen, da die Anforderungen aller Branchen und Länder berücksichtigt werden mussten. Sie haben sich daher in der Praxis für die meisten Anwendungsfälle als zu komplex für einen sinnvollen Einsatz erwiesen. Alternativ etablierten sich branchenspezifische Untermengen der Segmente und Elemente, so genannte *Subsets*, bei denen der exakte Inhalt der Nachricht wiederum zwischen den Unternehmen abgestimmt werden muss. Bei dieser Vorgehensweise existiert de facto ebenfalls keine semantische Interoperabilität, da die globalen Standardnachrichten zu komplex, und ihre Subsets wiederum branchenspezifisch sind. Beide Vorgehensweisen verbindet, dass die ausgetauschten Nachrichten nach dem Empfang mit hohem Aufwand auf die lokalen Systeme der Unternehmen gemappt werden müssen, da

²² Vgl. Janiesch; Thomas (2006), S. 63.

²³ Vgl. Mehnen (1989), S. 25.

²⁴ Vgl. Mehnen (1989), S. 44.

²⁵ Vgl. Mehnen (1989), S. 48.

das Format von EDIFACT-Nachrichten in den meisten Fällen nicht unmittelbar übertragbar ist. Zusammenfassend ist zu konstatieren, dass die *traditionellen Ansätze keine bzw. nur ungenügende syntaktische und semantische Interoperabilität ermöglichen* (vgl. Abb. 1).

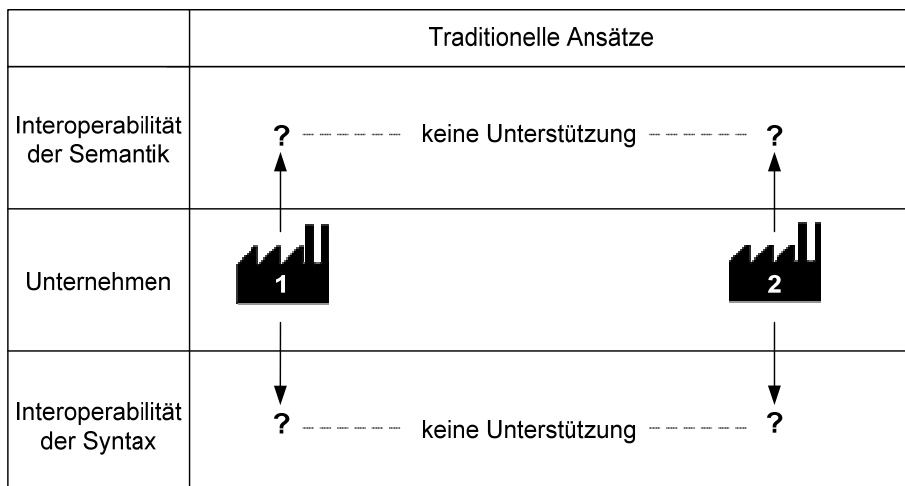


Abb. 1: Interoperabilität traditioneller Ansätze auf syntaktischer und semantischer Ebene

Die Nachteile der traditionellen Ansätze und die Ausbreitung eines einheitlichen Syntaxformats im *World Wide Web (WWW)* führten zu der Entwicklung neuer Standards, die die skizzierten Defizite beheben sollten.

2.3 Aktuelle Ansätze

2.3.1 Ausgangssituation

Die Grundlage für die Entwicklung neuer Standards bildete die Entstehung und Ausbreitung von XML als „Lingua Franca“ im Internet.²⁶ Die Entwicklung von XML hat ihren Ursprung in der Ausbreitung der *Hypertext Markup Language (HTML)* und dem damit verbundenen kommerziellen Erfolg des WWW.²⁷ Dieser Erfolg offenbarte jedoch zugleich die Grenzen von HTML, da sie sich aufgrund ihrer reinen Darstellungsfunktion für die steigenden Anforderungen webbasierter Anwendungen als ungeeignet erwies.²⁸ Dagegen existierte mit der *Standard Generalized Markup Language (SGML)* eine ausdrucksstarke Sprache für die Definition, Identifikation und Nutzung von Struktur und Inhalt eines Do-

²⁶ Vgl. Weitzel; Harder; Buxmann (2001), S. 3

²⁷ Vgl. Weitzel; Harder; Buxmann (2001), S. 15 f. sowie Wittenbrink et al. (2003), S. 17.

²⁸ Die unter der Bezeichnung „HTML-Dilemma“ subsumierten Defizite von HTML können z. B. Bosak (1997) entnommen werden.

kuments.²⁹ Die SGML war jedoch so komplex, dass die Anwendungsentwicklung auf ihrer Basis sehr zeit- und kostenaufwändig war.³⁰ Das Ziel von XML war es, eine für das Internet geeignete Sprache zu etablieren, die die Funktionalität von SGML für das Web bietet, ohne dessen Komplexität zu erreichen.³¹ Bedeutende Eigenschaften von XML-Dateien sind Dokumentenorientierung und hierarchischer Aufbau.³² Kurz nach der Veröffentlichung von XML entstand eine große Anzahl computergestützter Werkzeuge, die die Schemaerstellung, das *Mapping* auf andere Datenstrukturen und die Integration in andere Softwaresysteme unterstützten. Die hohe Verbreitung von XML förderte die Hoffnung, das Problem der Dateninteroperabilität durch ein einheitliches Syntaxformat zu lösen. In vielen Initiativen erfolgte eine rasche Integration von XML in die jeweiligen E-Business-Frameworks. Ein etablierter Standard auf Basis von XML ist RosettaNet.

2.3.2 RosettaNet

RosettaNet ist ein *Non-Profit*-Industriekonsortium mit mehr als 500 Unternehmen aus den Bereichen Halbleitertechnologie, elektronische Komponenten, Computertechnologie, Unterhaltungselektronik, Telekommunikation und Logistik.³³ Der von diesem Konsortium entwickelte Standard verfolgt das Ziel, integrierte Geschäftsprozesse entlang der *Supply-Chain* eines Unternehmens durch verringerte Durchlaufzeiten sowie höhere Effizienz und Zuverlässigkeit zu verbessern. Zudem soll die Zusammenarbeit und Kommunikation zwischen den beteiligten Unternehmen gefördert werden.³⁴ Den Hauptunterschied zwischen traditionellen Ansätzen wie EDIFACT und RosettaNet bildet dabei die Perspektive: Bei EDIFACT dominiert aufgrund des reinen Austauschs elektronischer Geschäftsdokumente die Datensicht, während bei RosettaNet die Prozesssicht im Fokus der Betrachtung steht. Der RosettaNet Standard umfasst folgende Komponenten:³⁵

- die *RosettaNet Standards Methodology* als Vorgehensmodell für die Unterstützung der Realisierung des Standards von der Planungs- bis zur Implementierungsphase.³⁶
- *Partner Interface Processes (PIP)* für die Definition von Prozessdialogen zwischen Geschäftspartnern.³⁷

²⁹ Weitzel; Harder; Buxmann (2001), S. 17.

³⁰ Vgl. Harold; Means (2005), S. 92

³¹ Vgl. Bray et al. (2006) sowie zu den Unterschieden zwischen XML und SGML vgl. Clark (1997).

³² Vgl. Wittenbrink et al. (2003), S. 598 sowie Ray (2001), S. 4.

³³ Vgl. RosettaNet (2006b) sowie Badakhchani (2004).

³⁴ Vgl. RosettaNet (2006a).

³⁵ Vgl. RosettaNet (2006f).

³⁶ Vgl. RosettaNet (2006g).

³⁷ Vgl. RosettaNet (2006c).

- *Dictionaries* für die Eigenschaften der PIP. Das *Business Dictionary* enthält die Eigenschaften der grundlegenden Geschäftsaktivitäten, während das *Technical Dictionary* die Definition von Produkten und Diensten unterstützt.³⁸
- Das *RosettaNet Implementation Framework (RNIF)* für die Spezifikation der Kapselung, Weiterleitung und des Transports der PIP-Nachrichten.³⁹

RosettaNet deckt drei Standardisierungsbereiche ab. Der erste Bereich umfasst die *Harmonisierung des verwendeten Vokabulars* durch die Dictionaries. Der zweite Bereich spezifiziert den *Austausch von Geschäftsnachrichten*. Dieser erfolgt mittels vordefinierter PIP, die Prozessdialoge zwischen den Geschäftspartnern enthalten. Der dritte Bereich beinhaltet das *Wrapping und den Transport der Nachrichten*, die das RNIF übernimmt.

Durch die starke Verbreitung von XML und dessen Nutzung durch RosettaNet ist die *syntaktische Interoperabilität zu anderen XML-basierten Standards relativ groß*. Da RosettaNet ein vertikaler Standard ist, also an die branchenspezifischen Bedürfnisse der partizipierenden Unternehmen angepasst wurde, und zudem auf den Bereich der Supply-Chain-Automatisierung und -optimierung fokussiert, ist semantische Interoperabilität nur partiell gegeben: Die Business Dictionaries inkl. des zugrunde liegenden Vokabulars stammen vornehmlich aus Industriebereichen, deren Unternehmen Mitglieder des Konsortiums sind.⁴⁰ Es existieren zwar Bestrebungen, RosettaNet in weiteren Branchen wie der chemischen oder der Erdöl verarbeitenden Industrie einzuführen, jedoch erweitern derartige Initiativen den Anwendungsbereich nur punktuell.⁴¹ Zudem erreicht RosettaNet aufgrund der relativ hohen Anfangsinvestitionen bei *kleinen und mittleren Unternehmen (KMU)* nur eine geringe Durchdringung.⁴² Bemühungen, horizontale Standards wie ebXML (vgl. Kap. 2.4.2) in RosettaNet zu integrieren, befinden sich noch in ihrem Anfangsstadium, so dass RosettaNet momentan nur *eingeschränkte semantische Interoperabilität besitzt* (vgl. Abb. 2).⁴³

³⁸ Vgl. RosettaNet (2006d).

³⁹ Vgl. RosettaNet (2006e) sowie RosettaNet (2001).

⁴⁰ Vgl. Badakhchani (2004).

⁴¹ Vgl. Jauhiainen et al. (2005), S. 43.

⁴² Vgl. Damodaran (2004), S. 194.

⁴³ Vgl. Jauhiainen et al. (2005), S. 43 sowie ebXML (2006b).

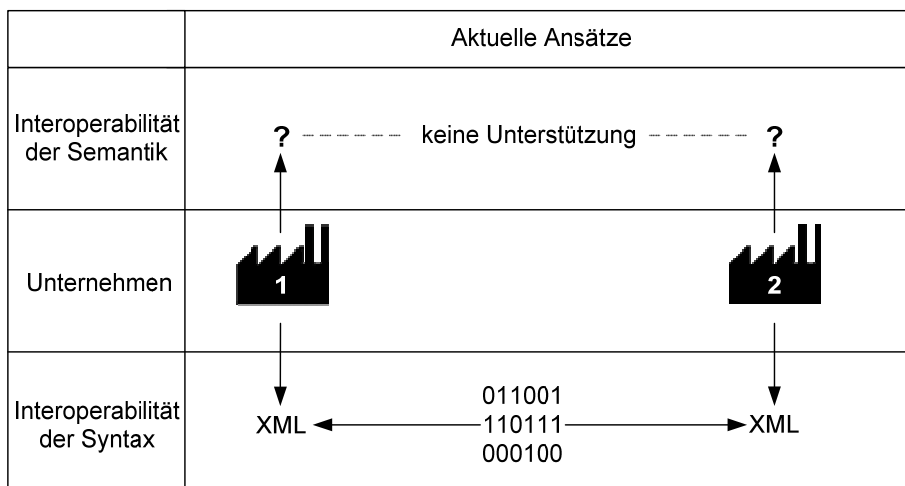


Abb. 2: Interoperabilität aktueller Ansätze auf syntaktischer und semantischer Ebene

An RosettaNet bleibt (stellvertretend für die aktuellen Ansätze) festzuhalten, dass sich die Hoffnung auf Interoperabilität durch die Verbreitung von XML als unbegründet herausstellte.⁴⁴ XML hat als Sprache nur die syntaktische Ebene vereinheitlicht. Mit den aktuellen Standards entstanden branchenspezifische Dialekte und Vokabulare, die die Struktur unterliegender Geschäftsdaten unterschiedlich modellieren und daher verschiedene Semantiken aufweisen. Dies führte in der Praxis dazu, dass Unternehmen, die unterschiedliche Standards nutzten, weiterhin ein Mapping der ausgetauschten Geschäftsdaten vorzunehmen hatten, um die semantischen Differenzen der jeweiligen Standards zu beseitigen. *Das ursprüngliche Problem aus den Anfangszeiten von EDI konnte somit nicht gelöst werden, sondern verschob sich nur auf eine andere Ebene.*

2.4 Ansätze der nächsten Generation

2.4.1 Ausgangssituation

Die mangelnde Harmonisierung der semantischen Ebene fördert die Gefahr einer unkontrollierten Ausbreitung individueller Strukturen zwischen den Marktteilnehmern. Um dies zu verhindern, entwickeln die UN und OASIS in der gemeinsamen Initiative ebXML eine Palette von Standards, um den elektronischen Handel von Unternehmen jeder Größe und jeder geographischen Region zu unterstützen.⁴⁵ Einer dieser Standards ist die Core Components Technical Specification, die eine Methode für die Entwicklung eindeutiger semantischer Einheiten und Aggregate enthält, auf deren Basis neue Geschäftsvokabulare erstellt und bereits bestehende Vokabulare restrukturiert werden können. Durch die eindeutige

⁴⁴ Weitere Ansätze sind bspw. xCBL sowie ältere Versionen von OAGIS.

⁴⁵ Vgl. ebXML (2006a). Eine Übersicht über die ebXML-Initiative befindet sich in Anhang A.

Definition atomistischer Datentypen, die jeder Komponente zugrunde liegen, stellt die CCTS die semantische Interoperabilität beim Austausch von Geschäftsdokumenten in Aussicht.

2.4.2 Core Components Technical Specification

Die Core Components Technical Specification wird inzwischen von einer Vielzahl führender Standards zur Beschreibung von Geschäftsdokumenten implementiert, da das Konzept die semantische Interoperabilität der Dokumente gewährleistet. Ziel der CCTS ist es insbesondere, Geschäftsdokumente so zu identifizieren, abzubilden und wiederzuverwenden, dass die Interoperabilität der Informationen bei der Nutzung in verschiedenen Geschäftskontexten maximiert wird.⁴⁶ Die CCTS-Methode berücksichtigt dabei die zwangsläufig anfallende Variabilität von Geschäftsdokumenten in verschiedenen Geschäftsszenarien und gewährleistet gleichzeitig ihre Kompatibilität. Der Standard selber ist implementierungsunabhängig, d. h. es werden keine Regeln über die zu verwendende Syntax vorgeschrieben. Die Methode unterliegt somit keinen maschinenspezifischen Einschränkungen und Anforderungen, die die Ausdrucksstärke der semantischen Information reduzieren könnten. Die CCTS beinhaltet folgende Kernaspekte:⁴⁷

- Logische Modellierung der Objekte: Die Blöcke werden, ähnlich wie bei Normalisierungsschemata von Datenbanken oder objektorientierten Ansätzen, so modelliert und zu größeren Blöcken aggregiert, dass Redundanzen so selten wie möglich auftreten.
- Namenskonventionen: eine Menge grammatikalischer Regeln, die auf dem *ISO Standard 11179 Part 5*⁴⁸ basieren. Einige dieser Regeln wurden in den Namenskonventionen modifiziert, um die Spezifika der Core Components zu adaptieren.
- Kontextspezifität: Geschäftsdokumente können so adaptiert werden, dass unternehmensindividuelle Charakteristika abgebildet werden. Dazu werden Kontextkategorien und -werte eingeführt, nach denen die Geschäftsdokumente differenziert werden.
- Syntaxneutralität: Die semantischen Modelle der Geschäftsdokumente werden implementierungsunabhängig beschrieben.

⁴⁶ Vgl. UN/CEFACT (2006b), S. 14.

⁴⁷ Vgl. Stuhec (2005), S. 7.

⁴⁸ Vgl. ISO/IEC (2005)

Das Modellierungskonzept von Geschäftsdaten nach CCTS ist zweistufig (vgl. Abb. 3):

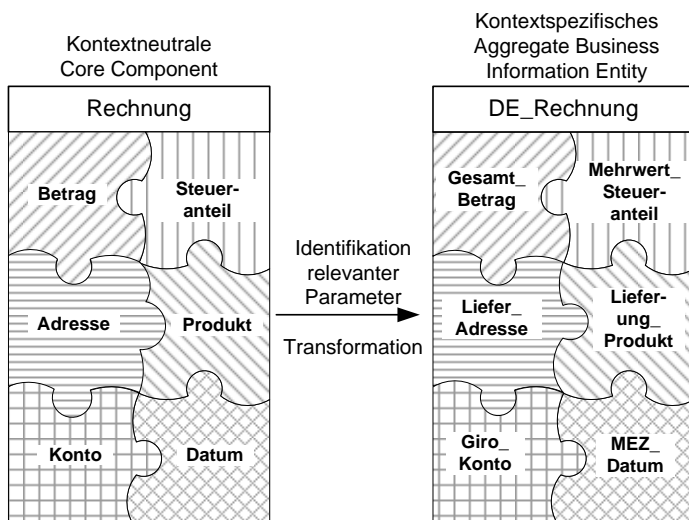


Abb. 3: Kontextspezifische Adaption von semantischen Blöcken

Auf der ersten Stufe werden *Core Components (CC)* spezifiziert, die semantische Einheiten und Blöcke bzgl. einer relevanten Geschäftsinformation generisch beschreiben.⁴⁹ Diese Blöcke sind zunächst kontextneutral und repräsentieren konzeptionelle Aspekte einer bestimmten Geschäftsinformation wie die logische Struktur und den Inhalt. Auf der zweiten Stufe werden diese Geschäftsdaten von der konzeptionellen Ebene auf einen realen Kontext übertragen. Dieser Prozess umfasst die Einordnung situationsbezogener Parameter in Kontextkategorien und die darauf basierende Adaption der Geschäftsdaten. Die kontextspezifischen Blöcke besitzen in der CCTS-Terminologie den Namen *Business Information Entity (BIE)*. Die Modellierung sowohl kontextneutraler als auch -spezifischer Geschäftsdaten ist vergleichbar zu der Grammatik einer menschlichen Sprache, da eine Geschäftsdatenentität drei Komponenten umfasst, die strukturell dem Aufbau eines Satzes gleichen:⁵⁰

- Die *Object Class (OC)* beinhaltet das Subjekt bzw. Objekt und repräsentiert die logische Gruppierung der Geschäftsdaten.
- Der *Property Term (PT)* enthält das Prädikat, das die Geschäftsinformation hinsichtlich ihrer Charakteristik differenziert.
- Der *Representation Term (RT)* als Adjektiv bzw. Adverb umfasst einen Datentyp, der den gültigen Wertebereich für die Geschäftsdaten bestimmt.

Diese drei Ausdrücke bilden den *Dictionary Entry Name (DEN)*, der in Repositories verwendet wird, um dort gespeicherte Geschäftsdaten aufzufinden und deren Wiederverwend-

⁴⁹ Vgl. Stuhec (2006a), S. 4.

⁵⁰ Vgl. Stuhec (2005), S. 12 sowie für die folgenden Aufzählungspunkte ebXML (2001d), S. 6.

barkeit zu fördern.⁵¹ Um die Methode der CCTS zu verinnerlichen, ist noch eine detaillierte Beschreibung dieses Grundprinzips erforderlich.

Core Components

Core Component ist der Oberbegriff für sämtliche Einheiten und Blöcke, die ein semantisch korrektes und sinnvolles Informationspaket beschreiben. Eine CC ist kontextneutral und enthält nur solche Informationen, die für die Beschreibung des spezifischen Pakets notwendig sind.⁵² Sie dient als Ausgangsmodell für eine kontextspezifische Transformation in ein Business Information Entity und beschreibt daher ein Objekt der realen Welt, das in jedem Kontext genutzt werden kann, wie z. B. ein Produkt, eine Adresse oder einen Betrag.⁵³ Es existieren drei Arten von Core Components: *Basic Core Components (BCC)*, *Aggregate Core Components (ACC)* und *Association Core Components (ASCC)*. Die kleinste Einheit mit semantischer Bedeutung wird Basic Core Component genannt. Jede BCC besitzt eine einzelne Geschäftscharakteristik, die in jedem Kontext verwendet werden kann. BCC können zu größeren semantischen Blöcken zusammengefasst werden, den Aggregate Core Components, deren *Object Class* sie übernehmen. Eine ACC beinhaltet eine Menge von Geschäftsinformationen, die zusammen ein einzelnes Objekt der Realwelt mit geschäftlicher Bedeutung ergeben. Sie sollte normalisiert sein, d. h. innerlich so zusammenhängend und äußerlich so abgegrenzt, dass ein eindeutiges, überschneidungsfreies Konzept beschrieben wird.⁵⁴ Aus Normalisierungsgründen folgt, dass eine umfangreiche ACC wie bspw. eine Adresse nicht nur BCC, sondern auch andere ACC wie bspw. das zugehörige Land, beinhaltet. Eine ACC, die in einer anderen ACC wiederverwendet wird, wird als Association Core Component bezeichnet. Zwischen ACC können zwei Arten von Assoziationsbeziehungen existieren, die Aggregation und die Komposition.⁵⁵

- Bei der Aggregation ist die assoziierte ACC zwar Teil der assoziierenden ACC, kann aber auch unabhängig von ihr existieren. Die Aggregation kann als Hierarchie verstanden werden, bei der ein Ganzes aus Teilen zusammengesetzt ist, die Teile jedoch auch für sich bestehen.
- Bei der Komposition wird die Existenz der assoziierten ACC durch die der assoziierenden ACC determiniert. Wird die assoziierende ACC gelöscht, so ist auch die assoziierte ACC nicht mehr existenzberechtigt und wird ebenfalls gelöscht. Die

⁵¹ Ein Beispiel für ein Repository kann ebXML (2001c) entnommen werden.

⁵² Vgl. UN/CEFACT (2006b), S. 83.

⁵³ Vgl. Stuhec (2006a), S. 6.

⁵⁴ Vgl. Stuhec (2006a), S. 6.

⁵⁵ Vgl. UN/CEFACT (2006b), S. 17.

Komposition ist somit eine strengere Form der Aggregation, bei der ebenfalls eine Hierarchie existiert, die Teile jedoch vom Ganzen existenzabhängig sind.

Innerhalb einer ACC sind sowohl BCC als auch ASCC Kardinalitäten für die unteren und oberen Schranken ihrer Auftretenshäufigkeit zuzuweisen. In Abb. 4 ist die ACC `Financial Account` dargestellt. Diese enthält eine Menge von BCC und ASCC, die zur Beschreibung des Konzepts „Konto“ erforderlich sind. Die enthaltenen Informationen der BCC sind erschließbar, wenn sie folgendermaßen gelesen werden: Die BCC `Financial Account.Account Name.Text` spezifiziert bspw. den Namen des Kontos in Form eines Textes, während die BCC `Financial Account.Currency.Code` einen Code für die Währung des Kontos (z. B. der ISO 4217) beinhaltet. Die ASCC `Financial Account.Owner.Party` wiederum enthält den Besitzer des Kontos, der durch eine natürliche oder juristische Person ausgedrückt wird.

<ACC>				
Financial Account. Details				
Object Class	Property Term	Representation Term	Type	Cardinality
Financial Account	Identification	Identifier	BCC	0..n
Financial Account	Type	Code	BCC	0..n
Financial Account	Account Name	Text	BCC	0..n
Financial Account	Currency	Code	BCC	0..n
Financial Account	Owner	Party	ASCC	0..n
Financial Account	Servicer	Party	ASCC	0..n
Financial Account	Information Rec.	Party	ASCC	0..n
Financial Account	Agent	Party	ASCC	0..n

Abb. 4: ACC `Financial Account` mit zugehörigen BCC und ASCC⁵⁶

Im Gegensatz zu ASCC, deren Representation Term durch die assoziierte ACC festgelegt ist, enthält der RT einer BCC einen *Core Data Type (CDT)*, der ihren Wertebereich spezifiziert.⁵⁷ *Da alle Core Components direkt oder indirekt auf CDTs basieren, bilden sie die Interoperabilitätsgrundlage in CCTS.* Daher müssen alle potenziellen CDTs vor ihrer Nutzung in einem zentralisierten Prozess geprüft werden. Gegenwärtig sind von der UN/CEFACT 22 verschiedene CDTs definiert und zur Verwendung freigegeben.⁵⁸ Ein CDT besteht aus genau einer *Content Component* und einer beliebigen Anzahl von *Supplementary Components*. Die Content Component ist ein Container für den aktuellen Wert, den ein CDT beinhaltet. Da die Information der Content Component isoliert betrach-

⁵⁶ Weitere Core Components können der aktuellen Core Component Library unter UN/CEFACT (2006a) entnommen werden.

⁵⁷ Vgl. UN/CEFACT (2006b), S. 17.

⁵⁸ Die Liste definierter CDTs des aktuellen Working Drafts kann UN/CEFACT (2006c) entnommen werden.

tet keine Bedeutung besitzt, sind in den Supplementary Components Metadaten zu deren Charakterisierung enthalten.⁵⁹ Content- und Supplementary Components basieren auf primitiven Datentypen wie z. B. *String* oder *Decimal* und erhalten innerhalb ihres CDTs Kardinalitäten. Eine schematische Darstellung der Aggregationsbeziehungen zwischen Core Components findet sich in Abb. 5.

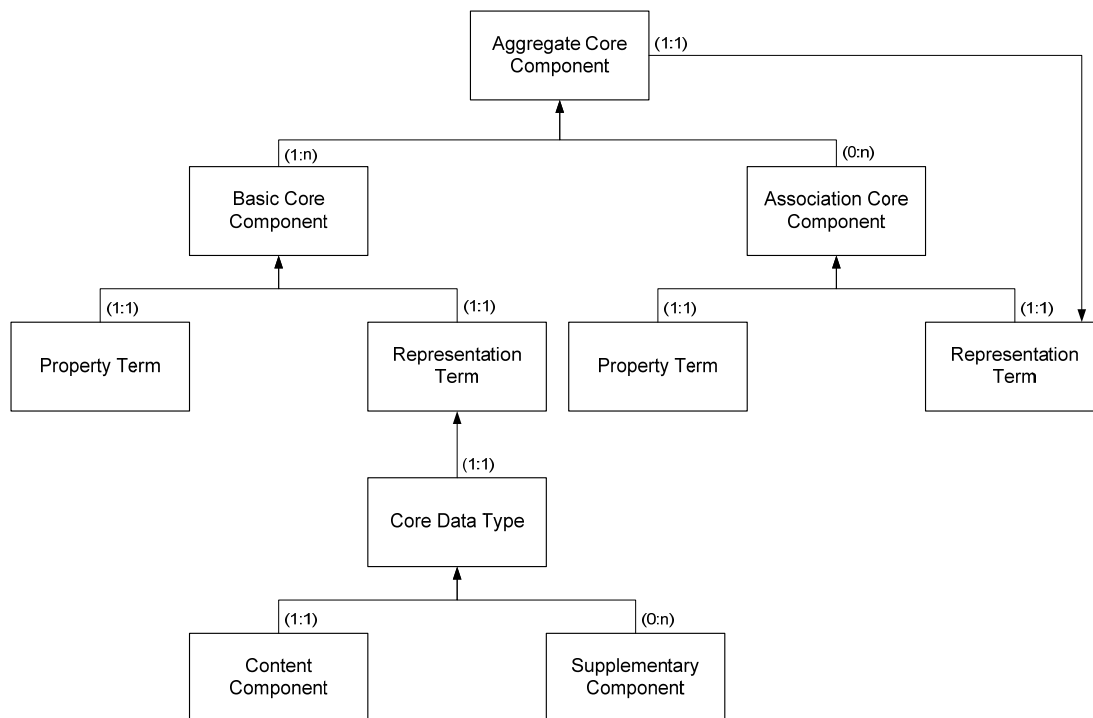


Abb. 5: Schematische Darstellung der Aggregationsbeziehungen zwischen Core Components

Business Information Entities

Der Oberbegriff Business Information Entity subsumiert Einheiten und Blöcke, die eine semantische Bedeutung in einem speziellen Geschäftskontext besitzen. BIE sind auf spezifische Geschäftssituationen adaptierte CC.⁶⁰ Zusätzlich zu der Object Class, den Kardinalitäten, dem Property Term und dem Representation Term, die auch CC aufweisen, können BIE einen *Qualifier Term (QT)* enthalten. Der QT ist ein Bezeichner, der die semantische Bedeutung des jeweiligen Kontexts präzisiert. Er kann sowohl einem Property Term als auch einer Object Class vorangestellt sein.⁶¹ Durch die Verwendung eines Bezeichners wird die Differenzierung verschiedener Domänen und Wertebereiche gefördert, da kontextspezifische BIE, die auf derselben CC basieren, durch den QT konkretisiert und unter-

⁵⁹ Vgl. Stuhec (2006b), S. 4.

⁶⁰ Vgl. UN/CEFACT (2006b), S. 18.

⁶¹ Vgl. UN/CEFACT (2006b), S. 47.

scheidbar werden.⁶² Analog zu den CC sind die BIE in drei Komponenten unterteilt: *Basic Business Information Entities (BBIE)*, *Aggregate Business Information Entities (ABIE)* und *Association Business Information Entities (ASBIE)*. Ein Basic Business Information Entity ist eine Basic Core Component, die an einen spezifischen Geschäftskontext adaptiert wurde.⁶³ Sie enthält eine einzelne Geschäftscharakteristik und repräsentiert ein Attribut innerhalb eines Aggregate Business Information Entities. Analog zu der Verhaltensweise der BCC übernehmen BBIE die Object Class des zugehörigen ABIE. Ein ABIE ist eine adaptierte ACC und repräsentiert ein Objekt der Realwelt in einem spezifischen Kontext.⁶⁴ Inhaltlich äußert sich die Transformation einer ACC zu einem ABIE in zweierlei Hinsicht:

- Bei der Adaption ist freigestellt, mit welchen Kardinalitäten BCC und ASCC der betreffenden ACC in das ABIE eingehen, solange es Teilmengen der Ausgangskardinalität sind.⁶⁵
- Die Datentypen der enthaltenen BBIE können mit Restriktionen belegt werden, die ihren Wertebereich einschränken.

Assoziationen auf kontextspezifischer Ebene werden durch Association Business Information Entities realisiert. Analog zu dem Konzept der ASCC können Aggregationen und Kompositionen formuliert werden. In Abb. 6 ist das ABIE *Credit_Financial Account* dargestellt, das aus der ACC *Financial Account* (vgl. Abb. 4) abgeleitet wurde.

<ABIE>						
Credit_Financial Account Details						
Object Class Qualifier	Object Class	Property Term Qualifier	Property Term	Representation Term	Type	Cardinality
Credit_	Financial Account	Business_	Identification	Identifier	BBIE	0..n
Credit_	Financial Account		Type	Code	BBIE	0..n
Credit_	Financial Account		Account Name	Text	BBIE	0..n
Credit_	Financial Account		Currency	Code	BBIE	0..n
Credit_	Financial Account	Primary_	Owner	Party	ASBIE	0..n
Credit_	Financial Account	Direct_	Servicer	Party	ASBIE	0..n

Abb. 6: ABIE *Credit_Financial Account*⁶⁶

⁶² Autoren wie Stuhec (2006a) verwenden eine CC innerhalb einer spezifischen Geschäftssituation sogar mehrfach, indem sie identische BIE mit unterschiedlichen QTs generieren. Eine derartige Vorgehensweise fördert jedoch Konsistenzprobleme, da die Summe der Kardinalitäten der qualifizierten BIE die Kardinalität der ursprünglichen CC nicht übersteigen darf. Von diesem Ansatz wird daher abstrahiert.

⁶³ Vgl. UN/CEFACT (2006b), S. 20.

⁶⁴ Vgl. UN/CEFACT (2006b), S. 19.

⁶⁵ Teilmenge kann auch Nichtexistenz eines Ursprungselements bedeuten, wenn die Ausgangskardinalität optional und die des BIE 0:0 ist.

⁶⁶ Vgl. UN/CEFACT (2006b), S. 20.

Durch die Adaption entfallen in dem ABIE die ASCC *Information Recipient* und *Agent*. Zudem werden der *Object Class* und einigen *Properties* der BBIE und ASBIE Bezeichner annotiert, um sie semantisch zu präzisieren. Die Beschränkung des Wertebereichs einzelner BBIE findet in dieser Darstellung keine Berücksichtigung. Sie wird durch das kontextspezifische Analogon zu *Core Data Types* realisiert, den *Business Data Types (BDT)*. Die *Content-* und *Supplementary Component* eines BDTs können durch Restriktionen in ihrem Wertebereich beschränkt werden und besitzen einen *Qualifier Term*, der sie semantisch präzisiert. Wird eine Restriktion formuliert, so ist ein *Restriction Type* und ein *Restriction Value* festzulegen, die Formulierung eines *Expression Types* und einer *Expression Type Language* ist optional. Der *Restriction Type* beinhaltet die Art der Typrestriktion (z. B. *Minimum Length* für einen *String*), der *Restriction Value* die Ausprägung dieser Restriktion (z. B. *3*).⁶⁷ Mit *Expression Types* können reguläre Ausdrücke definiert werden (z. B. für die Einschränkung des Alphabets) die *Expression Type Language* spezifiziert die Sprache dieses Ausdrucks (z.B. *Java* oder *Perl*).⁶⁸ Dem obigen Beispiel folgend sei angenommen, dass der Name des Bankkontos in einer alten Datenbank abgespeichert wird, die für *Strings* nur eine maximale Länge von zehn Zeichen bereitstellt. Dieser Kontext kann so modelliert werden, dass der *Representation Term* des Bankkontonamens einen Bezeichner erhält, und diesem modifizierten BDT Restriktionen annotiert werden können (vgl. Abb. 7). Sowohl die *Content-* als auch die *Supplementary Component* besitzen einen *Restriction Type*, der die maximale Länge des zugehörigen Datentyps *String* festlegt, und einen *Restriction Value*, der diesen auf maximal 10 Zeichen beschränkt.

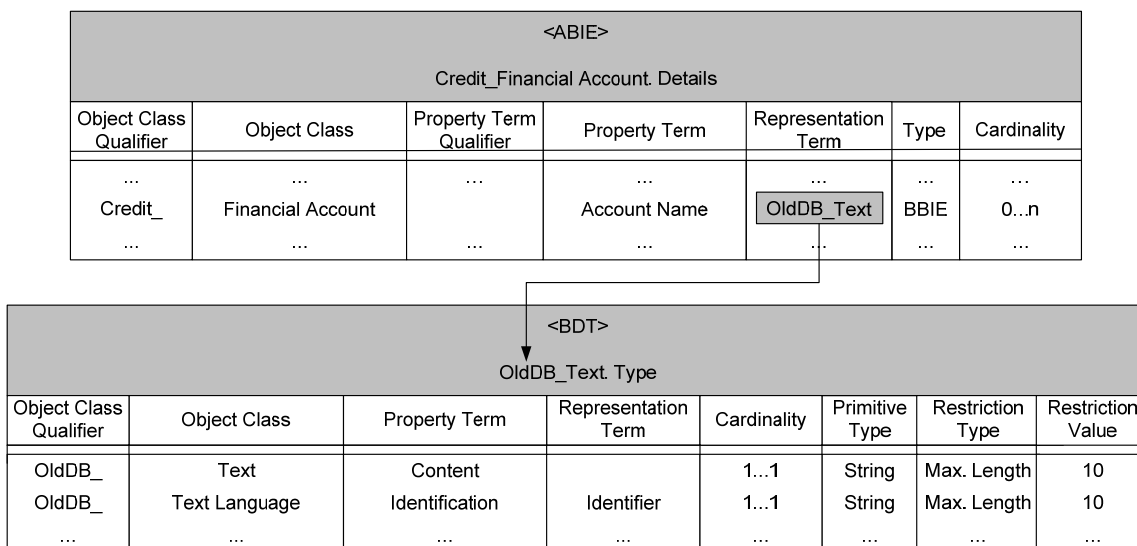


Abb. 7: Business Data Type mit annotierten Restriktionen

⁶⁷ Erlaubte Restriktionen an die primitiven Datentypen der *Content Component* und der *Supplementary Component* sind UN/CEFACT (2006c), S. 8 ff. zu entnehmen.

⁶⁸ Vgl. UN/CEFACT (2006b), S. 56 sowie Stuhc (2006b), S. 6.

Eine schematische Darstellung der Aggregationsbeziehungen zwischen Business Information Entities findet sich in Abb. 8.

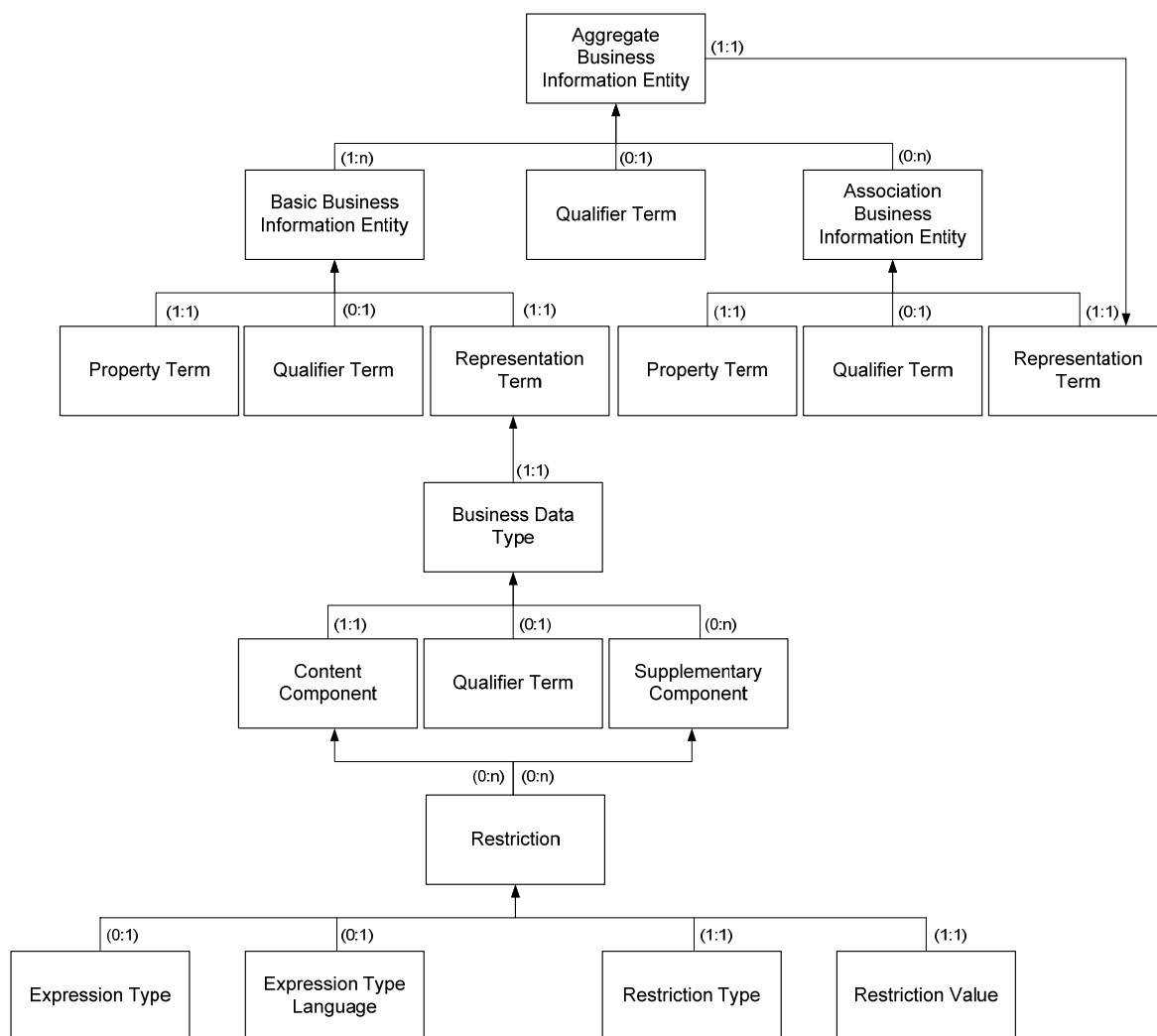


Abb. 8: Schematische Darstellung der Aggregationsbeziehungen zwischen Business Information Entities

Kontextprinzip und -kategorien

Jede Transformation einer Core Component in ein Business Information Entity erfolgt unter Einbezug eines geschäftsspezifischen Kontexts. Dieser Kontext repräsentiert die Umgebungsvariablen für den Austausch von Geschäftsinformationen zwischen Unternehmen. In der CCTS werden die individuellen Geschäftsparameter eines Unternehmens in acht verschiedene Dimensionen kategorisiert. Die Kategorien unterteilen sich in *Business Process*, *Product Classification*, *Industry Classification*, *Geopolitical Context*, *Official Constraints*, *Business Process Role*, *Supporting Role* sowie *System Capabilities*.⁶⁹ *Constraints*

⁶⁹ Für eine ausführlichere Beschreibung der Kategorien vgl. ebXML (2001b).

mit den Transformationsregeln von Core Components in Business Information Entities werden an die Ausprägungen innerhalb dieser Kategorien gebunden.⁷⁰

Kontextkategorie	Beschreibung	Externe Dokumente
Business Process	Geschäftsprozesse eines Unternehmens und die darin subsumierten Aktivitäten.	Konsistenzsicherung der Klassifizierung der Kontexte durch <i>UN/CEFACT Catalogue of Common Business Processes</i> ⁷¹
Product Classification	Produkte bzw. Dienstleistungen, die während eines Geschäftsprozesses ausgetauscht bzw. modifiziert werden.	Vereinheitlichung durch externe Codelisten für Produktklassifikationen wie z. B. die <i>Universal Standard Product and Service Specification</i> ⁷²
Industry Classification	Industrie bzw. Industriezweig, in dem der Geschäftsprozess stattfindet.	Spezifikation der Industrie durch externe Codelisten wie z. B. die <i>International Standard Industrial Classification</i>
Geopolitical Context	geographisch bedingte und kulturelle Faktoren sowie regionale, nationale und internationale Aspekte.	ISO 3166.1 sowie 3166.2
Official Constraints	durch gesetzliche oder regulatorische Anforderungen affektierte Aspekte einer Geschäftssituation	
Business Process Role Context	Rollenspezifische Faktoren innerhalb eines Geschäftsprozesses	Zu verwendende Rollen sind dem <i>UN/CEFACT Catalogue of Common Business Processes</i> zu entnehmen ⁷³
Supporting Role Context	Parteien, die an dem Geschäftsprozess beteiligt, aber nicht aktiv in ihn involviert sind	Ausprägungen sind einer standardisierten Klassifikation zu entnehmen, so z. B. der <i>UN/CEFACT Code List for DE 3035 Party Roles</i> ⁷⁴
System Capabilities Context	Systeme, die einen Einfluss auf die Geschäftssituation ausüben	

Tab. 4: Kontextkategorien der CCTS

Variantenerstellung mit Repositorydiensten

Um die kontextgetriebene Adaption von Core Components in Business Information Entities zu ermöglichen, werden in der CCTS Anforderungen an eine unterliegende Infrastruktur formuliert. Für die Gewährleistung von Interoperabilität müssen Business Information Entities auf einer einheitlichen semantischen Definition ihrer Core Components basieren.

⁷⁰ Vgl. UN/CEFACT (2006b), S. 68.

⁷¹ Vgl. UN/CEFACT (2006b), S. 69 sowie für den Catalogue of Common Business Processes ebXML (2001a).

⁷² Vgl. UN/CEFACT (2006b), S. 69 f.

⁷³ Vgl. ebXML (2001a), S. 54 f.

⁷⁴ UN/CEFACT (2006b), S. 72.

Diese wird durch die von der UN/CEFACT im Jahr 2006 veröffentlichte *Core Component Library (CCL)* bereitgestellt.⁷⁵ Core Data Types, die die Datentypen der CC spezifizieren, sind ebenfalls öffentlich verfügbar. Des Weiteren fordert die CCTS ein Repository, um CC, BIE, Kontextausprägungen und die Beziehungen dieser Komponenten untereinander zu speichern und mit Metadaten zu versehen.⁷⁶ Das Repository ist die zentrale Anlaufstelle für Nutzer, die den Austausch ihrer Geschäftsdokumente der CCTS entsprechend gestalten wollen. Mit der *UN/CEFACT Modeling Methodology (UMM)* existiert eine Modellierungsmethode, um Geschäftsprozesse der Nutzer zu analysieren und relevante Kontexte sowie potenzielle Aggregate Business Information Entities zu identifizieren.⁷⁷ Die potenziellen ABIE werden innerhalb des Repositories auf Ähnlichkeiten zu bereits vorhandenen ABIE verglichen und diesem gegebenenfalls hinzugefügt (vgl. den Übernahmeprozess in Anhang B). Durch die sukzessive Addition von BIE und Kontextausprägungen steigt die Menge kontextspezifischer Daten innerhalb des Repositories. Durch die Analyse der gemeinsamen Eigenschaften der ABIE kann zudem die Struktur unterliegender ACC identifiziert und im Repository gespeichert werden.⁷⁸ Die Relationen zwischen BIE, Kontexten und CC sind dabei durch eine *Constraint Language* zu beschreiben.⁷⁹ Der adressierbare Nutzerkreis des Repositories steigt mit der Menge der verfügbaren CC, BIE und Kontexten. *Zum Zeitpunkt dieser Arbeit existiert zwar die CCL für Core Components. Ein Repository mit ABIE und Kontexten konnte jedoch nicht identifiziert werden.* Da CCTS ein implementierungsunabhängiger Standard ist, und der Fokus auf der Betrachtung semantischer Modelle liegt, sollten die Komponenten innerhalb eines Repositories zunächst syntaxneutral formuliert sein (vgl. im Folgenden Abb. 9). Das syntaktische *Binding* an ein spezifisches Format wird erst vorgenommen, wenn die Komponenten außerhalb des Repositories verwendet werden.⁸⁰ Letzteres kann bspw. durch die Modellierung von CCTS-Komponenten als UML-Modelle erfolgen, die durch computergestützte Werkzeuge in XML-Schemata überführt werden.⁸¹ Trotz der Tatsache, dass CCTS prinzipiell in anderen Syntaxformaten realisierbar ist, basieren die aktuellen Implementierungen zumeist auf XML. Dieser Trend erscheint durch die Entstehung der CCTS aus der ebXML-Initiative und der starken Verbreitung von XML durchaus nachvollziehbar.

⁷⁵ Die CCL ist in Form eines Excel-Spreadsheets unter UN/CEFACT (2006a) abrufbar.

⁷⁶ Vgl. UN/CEFACT (2006b), S. 90-106.

⁷⁷ Auf eine detaillierte Beschreibung der UMM wird aus Platzgründen verzichtet, ist aber bei UN/CEFACT (2003b) nachzulesen.

⁷⁸ Vgl. UN/CEFACT (2006b), S. 105 f.

⁷⁹ Vgl. UN/CEFACT (2003a), S. 59.

⁸⁰ Vgl. UN/CEFACT (2006b), S. 92.

⁸¹ Vgl. Stuhec; Crawford (2006), S. 7 sowie für die UML-Modelle UN/CEFACT (2006d).

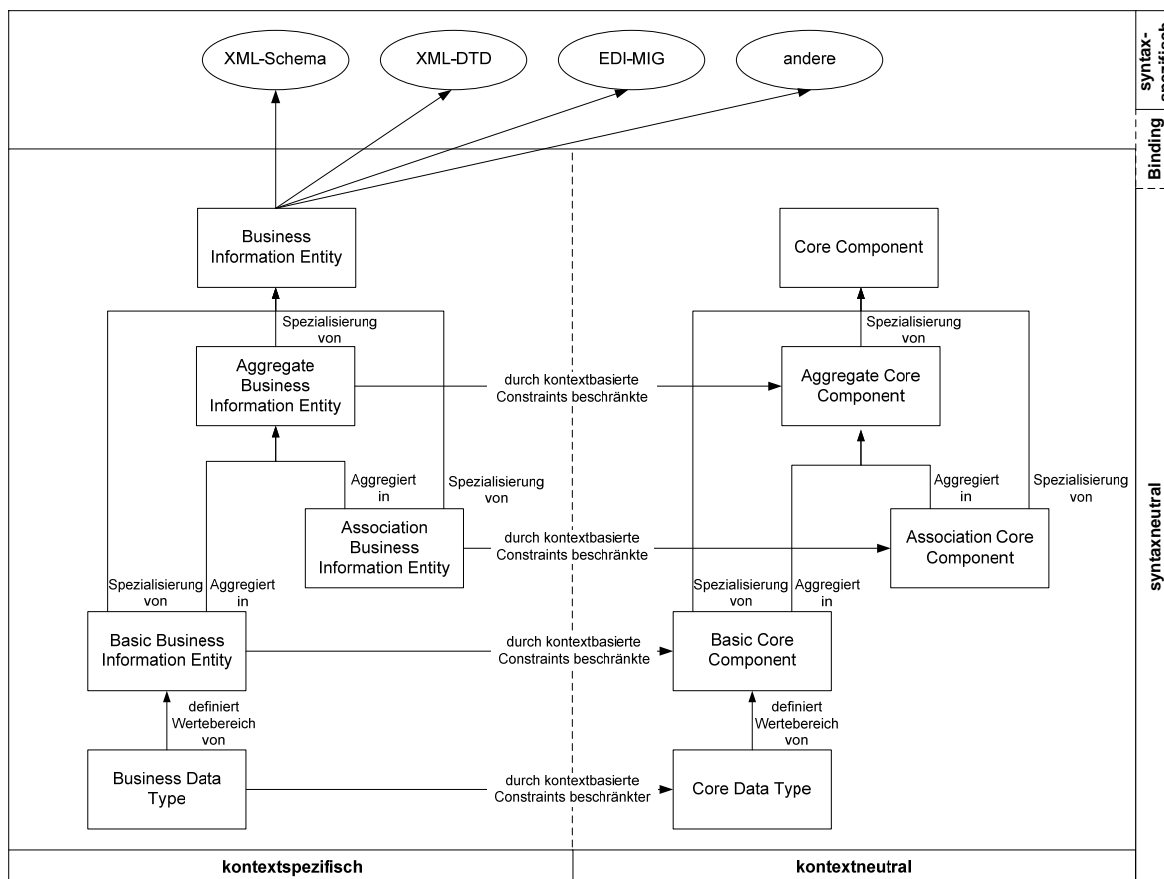


Abb. 9: Übergeordneter Zusammenhang der CCTS-Komponenten

Da die CCTS als Metastandard formuliert wurde, ist sie vor ihrem Einsatz in der Praxis zu implementieren. Einen *Early Adopter* der CCTS stellt die *Universal Business Language* dar.⁸²

2.4.3 Universal Business Language

Die Universal Business Language ist eine kostenfreie Bibliothek standardisierter, elektronischer und XML-basierter Geschäftsdokumente der OASIS.⁸³ Die Struktur der Geschäftsdokumente basiert auf der CCTS, eigenen Angaben zufolge war UBL sogar der erste Standard, der die CCTS tatsächlich implementiert hat.⁸⁴ Die erste Version wurde nach dreijähriger Entwicklungszeit im November 2004 veröffentlicht und enthielt neun Dokumententypen, die den Beschaffungsprozess von der Bestellung bis zur Faktura unterstützen. Die zweite Version erweitert den Anwendungsbereich um 23 zusätzliche Dokumententypen.⁸⁵ Historisch betrachtet entstammt UBL der ebXML-Initiative, da dort die Entwicklung eines

⁸² Vgl. OASIS (2007), Punkt 13.

⁸³ Vgl. OASIS (2007), Punkt 1.

⁸⁴ Vgl. OASIS (2007), Punkt 13.

⁸⁵ Vgl. OASIS (2007), Punkt 17 sowie OASIS (2006) Punkt 4.

standardisierten, XML-basierten Geschäftsdatenformats für ebXML gefordert wurde.⁸⁶ Die Basis dieser Arbeit bildete eine bereits existierende Sammlung von XML-Schemata für den B2B E-Commerce, die *xCBL (XML Common Business Library)* von Commerce-One und SAP.⁸⁷

Das Entwurfsziel von UBL bestand in einer 80:20-Lösung. Es sollten die zwanzig Prozent der Geschäftsdokumente realisiert werden, die für achtzig Prozent der Geschäftstransaktionen zwischen Unternehmen verantwortlich sind.⁸⁸ Diese Entscheidung impliziert, dass selten benötigte Dokumente nicht in UBL integriert wurden, bzw. implementierte Dokumente bei der Kardinalität mehr Gestaltungsspielraum erhielten als in gewissen Kontexten erlaubt. UBL ermöglicht jedoch zusätzlich eine kontextspezifische Adaption der implementierten Geschäftsdokumente bzw. die Erstellung individueller Geschäftsdokumente durch eine strukturierte Vorgehensweise.⁸⁹

Inhaltlich umfasst UBL eine Bibliothek wiederverwendbarer Datenelemente wie z. B. Adresse oder Artikel.⁹⁰ Diese *UBL-Library* basiert auf konzeptionellen Modellen in Form von Business Information Entities, wie sie in der CCTS spezifiziert sind. Die Komponenten der UBL-Library werden durch eine Sammlung von XML-Schemata gängiger Geschäftsdokumente, wie z. B. Bestellung oder Rechnung, ergänzt (vgl. die Darstellung aus *Altova XMLSpy® 2006* in Abb. 10). Die Dokumente bestehen aus Komponenten der UBL-Library und sind in einen spezifischen Geschäftsprozess eingebettet. Dort werden sie eingesetzt, wenn Informationen zwischen den am Prozess beteiligten Akteuren bzw. Rollen ausgetauscht werden. Ein exemplarischer UBL-Prozess wie die Bestellung ist in Abb. 11 modelliert.

⁸⁶ Vgl. OASIS (2007), Punkt 12.

⁸⁷ Vgl. UBL Marketing Subcommittee (2002), S. 6 sowie xCBL (2007).

⁸⁸ Vgl. Maler (2003), S. 32 sowie Wittenbrink et al. (2003), S. 607.

⁸⁹ Vgl. Taschek (2003), S. 1 sowie OASIS (2006) Punkt 4.

⁹⁰ Vgl. OASIS (2006), Punkt 1.

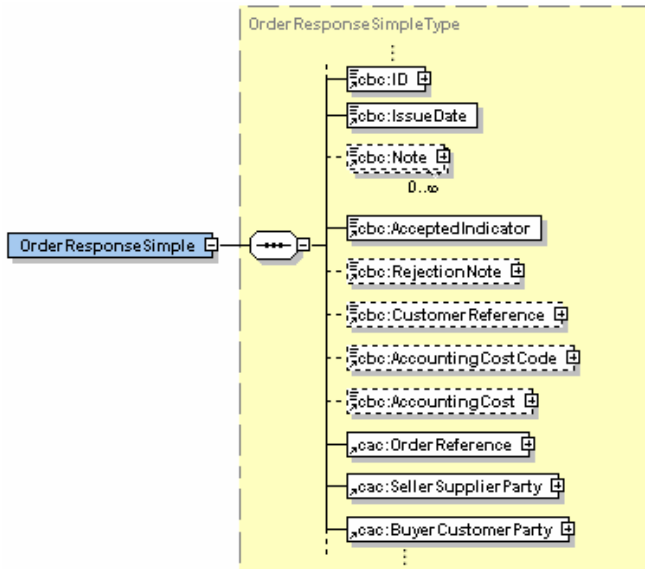


Abb. 10: UBL-Geschäftsdokument Order Response Simple⁹¹

Für die kontextspezifische Adaption der Geschäftsdokumente wurden die Kontextkategorien der CCTS übernommen. Diese sind mit ihren Ausprägungen in den Varianten zu speichern, um eine Rückverfolgung auf das ursprüngliche Geschäftsdokument zu ermöglichen und so eine branchen- und länderübergreifende Interoperabilität zu gewährleisten.

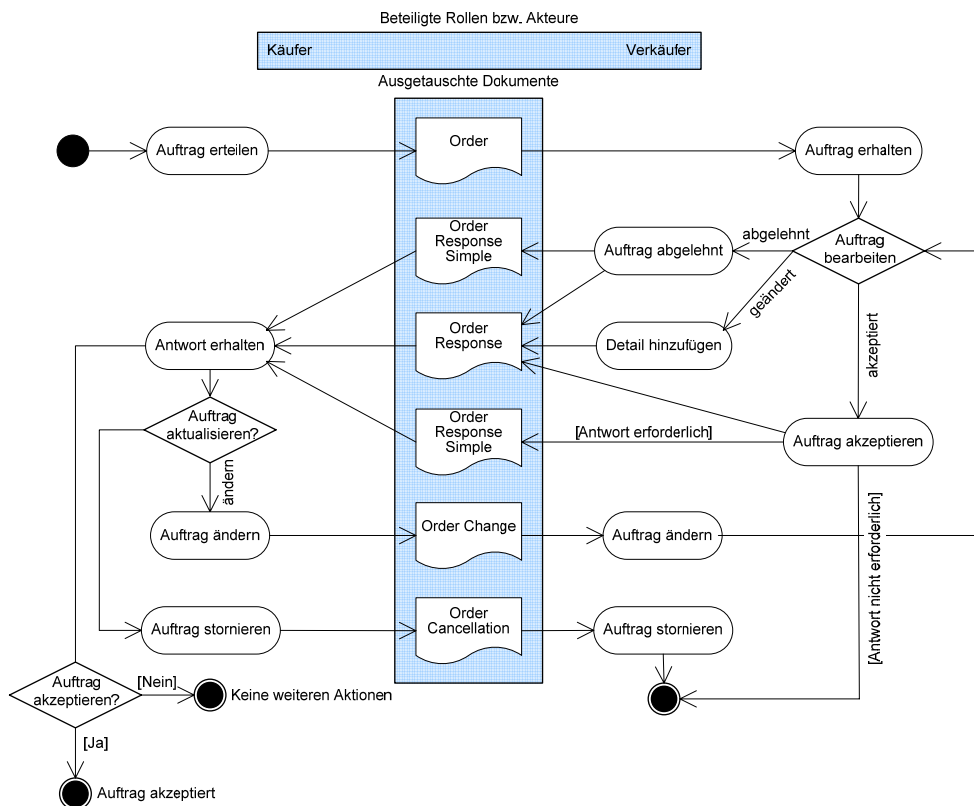


Abb. 11: Bestellprozess in UBL⁹²

Variantenerstellung und Modellierung individueller Geschäftsdokumente

Mit dem skizzierten 80:20-Prinzip versucht UBL, den offenkundigen Fehler bisheriger Standards zu vermeiden, eine allumfassende Bibliothek von Geschäftsdokumenten zu definieren. Da die Anforderungen in verschiedenen Unternehmen, Branchen und Ländern divergieren ist in gewissen Fällen eine Adaption von Geschäftsdokumenten erforderlich.⁹³ Die individuellen Anwendungssituationen werden in UBL durch das in der CCTS spezifizierte Kontextprinzip abgebildet.⁹⁴

Geschäftsdokumentvarianten werden auf technischer Ebene durch die Ableitung aus einem UBL-Ausgangsschema realisiert.⁹⁵ In Übereinstimmung mit dem Ableitungsmechanismus von *XML-Schema-Definition-Dateien (XSD)*⁹⁶ der W3C ermöglicht UBL sowohl die *Einschränkung als auch die Erweiterung von Geschäftsdokumenten in spezifischen Kontexten*.⁹⁷ Um eine UBL-kompatible Dokumentenvariante zu erzeugen, sind bei der Generierung spezielle Regeln einzuhalten. Zudem ist es möglich, aus der UBL-Library individuelle Geschäftsdokumente und UBL-inkompatible Varianten zu erstellen, die durch ihren Aufbau aus Business Information Entities und Data Types zumindest auf Komponentenebene interoperabel sind. Der Unterschied zwischen Kompatibilität und Interoperabilität besteht darin, dass ein interoperables Dokument zwar syntaktisch und semantisch präzisiert ist, jedoch nicht wie ein kompatibles Dokument durch einen XSD-Prozessor automatisiert auf sein Ausgangsdokument zurückgeführt werden kann.

Erstellung UBL-kompatibler Varianten

Eine UBL-Komponente kann in zweierlei Hinsicht modifiziert werden: durch Spezifikation einer Einschränkung und durch Erweiterung der Komponente. Die Funktionsweise dieser Modifikationen wird im Folgenden anhand eines Beispiels, des UBL-Aggregate Business Information Entity `Financial Account` verdeutlicht (vgl. Abb. 12).

⁹¹ Aus Gründen der Übersichtlichkeit wird von *Annotations* abstrahiert.

⁹² In Anlehnung an OASIS (2006), Punkt 4.

⁹³ Vgl. OASIS (2004), Punkt 1.

⁹⁴ Den folgenden Ausführungen liegen die *Guidelines for the Customization of UBL v1.0 Schemas* zugrunde, da für UBL 2.0 zum aktuellen Zeitpunkt kein entsprechendes Dokument verfügbar war.

⁹⁵ Vgl. OASIS (2004), Punkt 1.

⁹⁶ Vgl. W3C (2004).

⁹⁷ Vgl. OASIS (2004), Punkt 1.


```

<xsd:complexType name="Financial Account">
  <xsd:sequence>
    <xsd:element ref="ID" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="Name" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="AccountTypeCode" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="CurrencyCode" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="PaymentNote" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="FinancialInstitutionBranch" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="Country" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

```

Abb. 12: UBL-Aggregate Business Information Entity Financial Account

In der Abbildung ist ein konzeptioneller Ausschnitt⁹⁸ einer XSD dargestellt, die das Aggregate Business Information Entity inklusive der Basic Business Information Entities und Association Business Information Entities ID, Name, AccountTypeCode, CurrencyCode, PaymentNote, FinancialInstitutionBranch und Country aus der UBL-Library visualisiert.⁹⁹ Mit der Ableitung von XSD-Dateien ist es möglich, konform zu den Regeln der CCTS Kardinalitäten von Komponenten *einzuschränken oder ggf. zu eliminieren* (wenn die Kardinalität von (m:n) auf (0:0) reduziert wird). Dabei ist jedoch nur die Bildung einer Teilmenge der Ausgangsmenge erlaubt.¹⁰⁰ Technisch wird dieser Vorgang durch den `<restriction>`-Tag realisiert. Abb. 13 führt das in Abb. 12 eingeführte Beispiel weiter: Es wird angenommen, dass ein Auktionskonto immer einen Zahlungsvermerk enthalten muss.

```

<xsd:complexType name="Auction_Financial Account">
  <xsd:restriction base="cat:Financial Account">
    <xsd:sequence>
      <xsd:element ref="ID" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="Name" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="AccountTypeCode" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="CurrencyCode" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="PaymentNote" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="FinancialInstitutionBranch" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="Country" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:restriction>
</xsd:complexType>

```

Abb. 13: Abgeleitetes UBL-Aggregate Business Information Entity Auction_Financial Account

Die Bildung einer Variante ohne PaymentNote wäre ebenfalls zulässig, da das Ausgangsschema die Kardinalität (0:n) aufweist.

⁹⁸ Aus Gründen der Übersichtlichkeit wird von Annotations abstrahiert.

⁹⁹ Vgl. OASIS UBL TC (2007).

¹⁰⁰ Vgl. OASIS (2004), Punkt 3.

Bei der *Erweiterung* werden dem UBL-Ausgangsschema Informationen hinzugefügt. Dieser Vorgang ist nicht konform zu der aktuellen Vorgehensweise bei der Variantengenerierung nach CCTS, da dort bei der kontextspezifischen Generierung von Business Information Entities ausschließlich Einschränkungen zulässig sind. In UBL existiert diese Möglichkeit. Dabei werden in der XSD mit dem `<extension>`-Tag am Ende des Ausgangsdokuments spezifische Informationen hinzugefügt. Bezogen auf das obige Beispiel wäre dies notwendig, wenn dem Konto eine weitere ID hinzugefügt würde, um es intern weiter zu verarbeiten (vgl. Abb. 14).

```
<xsd:complexType name="Internal_Financial Account">
  <xsd:extension base="cat:Financial Account">
    <xsd:element ref="Internal_ID" minOccurs="1" maxOccurs="1"/>
  </xsd:extension>
</xsd:complexType>
```

Abb. 14: Erweitertes UBL-Aggregate Business Information Entity `Internal_Financial Account`

Erstellung UBL-inkompatibler Varianten und individueller Geschäftsdokumente

Der Bedarf an UBL-inkompatiblen Varianten ist durch zwei Anforderungen aus der Praxis determiniert. Einerseits existieren Szenarien, in denen *Modifikationen an dem UBL-Schema notwendig sind, die über die zulässigen Möglichkeiten der XSD-Einschränkung hinausgehen*, wie bspw. die Eliminierung von Elementen, die im Ausgangsschema als notwendig gekennzeichnet sind.¹⁰¹ Die zweite Anforderung besteht in der Addition eines Elementes an einer bestimmten Stelle innerhalb des Dokuments. Diese Anforderung ist bei der XSD-Erweiterung nicht zulässig, da dort das Element mit dem `<extension>`-Tag nur am Ende der Sequenz hinzugefügt werden kann.¹⁰² Um diese Anforderungen mit der technischen Komponente der XSD-Ableitung in Einklang zu bringen, werden in UBL sog. *Ur-Schemata* eingeführt. Ein Ur-Schema ist ein hypothetisches Schema hinter dem originären UBL-Schema, bei dem alle Elemente optional und abstrakt¹⁰³ sind.¹⁰⁴ Durch die XSD-Ableitung aus dem Ur-Typ können Einschränkungen formuliert werden, die mit dem originären UBL-Schema unmöglich wären. Ein solches Dokument ist jedoch nicht mehr kompatibel zu dem Ausgangsschema, so dass ein XSD-Prozessor die wechselseitige Beziehung nicht erkennen kann. Durch den standardisierten Prozess der Ableitung von Ur-Typen ist jedoch zumindest die Interoperabilität der Geschäftsdokumente sichergestellt.

¹⁰¹ Vgl. OASIS (2004), Punkt 3.

¹⁰² Vgl. OASIS (2004), Punkt 4.

¹⁰³ Abstrakt bedeutet erst durch Ableitung verwendbar.

¹⁰⁴ Vgl. OASIS (2004), Punkt 4.

Die *Modellierung individueller Geschäftsdokumente* in UBL erfolgt nach einer strukturier-ten Vorgehensweise.¹⁰⁵ Eignet sich keines der in der UBL-Library definierten Dokumente oder zugehörigen Ur-Schemata für die Adaption an einen bestimmten Kontext, so können individuelle Geschäftsdokumente auf Basis der Datentypen und Business Information Enti-ties von UBL definiert werden.¹⁰⁶ In folgendem Beispiel sei angenommen, ein Unterneh-men benötigte eine spezifische Produktbeschreibung innerhalb eines Geschäftsdokuments. Diese Beschreibung enthalte eine ID, einen Namen und eine Lagerkapazität. Die Definiti-on gestaltete sich entsprechend Abb. 15.

```
<xsd:complexType name="ProductDescriptionType">
  <xsd:sequence>
    <xsd:element name="ID" type="cct:IdentifierType"/>
    <xsd:element name="Name" type="cct:NameType"/>
    <xsd:element name="Capacity" type="cct:AmountType"/>
  </xsd:sequence>
</xsd:complexType>
```

Abb. 15: Modellierter Typ ProductDescriptionType

Grundsätzlich sollte die Bildung einer Variante aus einem ABIE stets einer Neuerstellung individueller Dokumente vorgezogen, um konform zu der UBL-Spezifikation zu modellie-ren und potenzielle semantische Redundanzen zu existierenden UBL-Dokumenten zu ver-meiden.¹⁰⁷

Dokumentation der Adaption

Jede Adaption eines UBL-Schemas ist grundsätzlich zu dokumentieren.¹⁰⁸ Inhaltlich sind die betreffenden Kontextkategorien und -ausprägungen festzuhalten, die zu der Generie-rung der jeweiligen Variante führten. Die Kontexte sind innerhalb eines *<Context>*-Tags festzuhalten, der wiederum innerhalb eines *<Contextualization>*-Tags gekapselt ist. In XML-Darstellung gestaltete sich diese Beziehung bspw. folgendermaßen (vgl. Abb. 16):

¹⁰⁵ Vgl. Crawford (2002), S. 46.

¹⁰⁶ Vgl. OASIS (2004), Punkt 4.

¹⁰⁷ Vgl. OASIS (2004), Punkt 6.

¹⁰⁸ Vgl. OASIS (2004), Punkt 3.

```

<xsd:annotation>
  <xsd:documentation>
    <ubl:Contextualization>
      <ubl:Context>
        <ubl:Geopolitical>Germany</ubl:Geopolitical>
        <ubl:IndustryClassification>Apparel</ubl:IndustryClassification>
        <ubl:ProductClassification>Shoes</ubl:ProductClassification>
      </ubl:Context>
    </ubl:Contextualization>
  </xsd:documentation>
</xsd:annotation>

```

Abb. 16: Dokumentation von Kontextkategorien und -werten

Prinzipiell können Geschäftsdokumente beliebig oft an einen Kontext adaptiert werden. Jedoch dürfen bereits spezifizierte Kontextausprägungen bei weiterer Adaption nur spezialisiert werden (Für weitere Adaptionen des Dokuments aus Abb. 16 könnten dies bspw. Regionen wie *North Rhine-Westphalia* sein, nicht jedoch Länder wie bspw. *France*).

2.4.4 Gegenüberstellung und Bewertung von CCTS und UBL

Eigenen Angaben zufolge ist UBL die erste Sprache, die CCTS implementiert. Der Vergleich beider Ansätze verdeutlicht jedoch signifikante Unterschiede. So basieren alle Dokumente in UBL auf Business Information Entities. Core Components, wie in der CCTS spezifiziert, existieren nicht. OASIS begründet diese Tatsache mit dem Argument, dass in UBL die Kontextkategorie *Business Process* bereits spezifiziert ist, und die Geschäftsdokumente darin eingebettet sind.¹⁰⁹ *Eine Variantenerstellung aus CC findet nicht statt.* In UBL können Business Information Entities wie in der CCTS auf qualifizierten und unqualifizierten Datentypen basieren, die in einer Library bereitgestellt sind. Momentan enthält das UBL-Schema qualifizierter Datentypen nur Modifikationen des Basistyps `Code`, und die einzelnen Typen enthalten auch keine Einschränkungen, sondern sind mit *Default*-Werten versehen. *Restriktionen von Datentypen in den Geschäftsdokumenten existieren bisher nur bei der Spezifikation vordefinierter Werte.* Ein weiterer Unterschied bei der Variantengenerierung ist die Möglichkeit, den *Variante* in UBL *individuelle Elemente mit dem <extension>-Tag hinzuzufügen.* In der CCTS sind dagegen nur Projektionen von CC zulässig. Zudem erfolgt die *Erstellung und Adaption von Dokumenten in UBL durch die Festlegung auf XML-Schemata syntaxspezifisch*, während in der CCTS eine syntaxneutrale Modellierung und Adaption gefordert wird, um die semantische Bedeutung der Dokumente losgelöst von ihrer syntaktischen Repräsentation zu betrachten. Einen weiteren bedeutsamen Unterschied stellt die *unterschiedliche Nutzung von Infrastrukturdiensten in UBL und CCTS* dar. Die CCTS baut durch ihre Zugehörigkeit zu der ebXML-Initiative auf

¹⁰⁹ Vgl. Crawford (2002), S. 40.

andere dort festgelegte Standards, wie z. B. Repositorydienste, auf. UBL dagegen ist nicht an eine spezifische Infrastruktur gebunden, um Anwendungsmöglichkeiten von der komplexen serviceorientierten Architektur bis zum einfachen Austausch der Dokumente via E-Mail zu unterstützen.¹¹⁰

Im Gegensatz zu aktuellen Ansätzen wie RosettaNet sind die CCTS bzw. die CCTS implementierenden Standards branchen- und länderübergreifend ausgerichtete, und somit vertikale Standards. Aktuelle Implementierungen von CCTS wie UBL basieren zumeist auf XML, wodurch syntaktische Interoperabilität gewährleistet ist. Die semantische Interoperabilität eines Geschäftsdokuments wird durch die Rückführung der kontextspezifischen Business Information Entities auf ihre ursprünglichen Core Components (bzw. auf ihre minimal adaptierten BIE in UBL) hergestellt.¹¹¹ Die semantische Bedeutung einzelner Komponenten ist dadurch eindeutig festgelegt. Zudem ist ein automatisiertes Mapping der ausgetauschten Geschäftsdokumente auf die internen Systeme der Unternehmen möglich. *Die CCTS bzw. die CCTS implementierenden Standards ermöglichen daher eine syntaktische und semantische Interoperabilität von Geschäftsdokumenten* (vgl. Abb. 17).

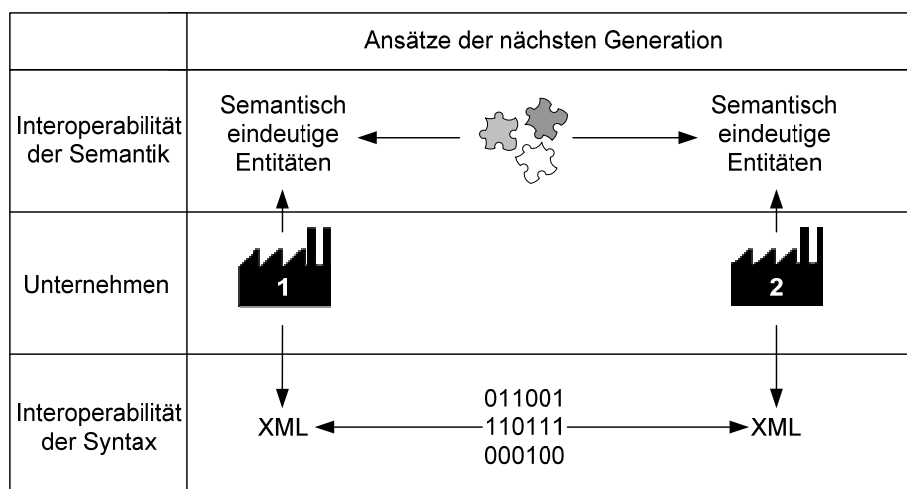


Abb. 17: Interoperabilität von Ansätzen der nächsten Generation auf syntaktischer und semantischer Ebene

Der Einsatz von CCTS bzw. UBL beinhaltet jedoch auch Probleme und Nachteile, die zu evaluieren sind.

Einen bedeutenden Nachteil stellt die willkürliche Ergänzung von Elementen zu Varianten mittels des <extension>-Tags dar. In der CCTS werden Varianten durch Einschränkung einer von einer Expertengruppe sorgfältig geprüften und definierten CC generiert. CC

¹¹⁰ Vgl. OASIS (2007), Punkt 12.

¹¹¹ Vgl. UN/CEFACT; OASIS (2001), S. 13

sind kontextunabhängig, normalisiert und hinsichtlich ihrer Semantik präzisiert. Die Konsistenz einer aus dieser CC generierten Variante ist somit zumindest grundlegend gesichert, da sie nur eine Teilmenge der Ausgangsmenge enthalten kann. Für UBL-Varianten ist dieser Sicherungsmechanismus nicht gegeben. *Frei erweiterte UBL-Varianten bergen somit die Gefahr der semantischen Inkonsistenz.* Diese kann auch in der CCTS auftreten, wenn Core Components weiterentwickelt werden: Die Definitionen eines ABIE wird innerhalb des Repositories zusammen mit ihrer Kontextausprägung und der Referenz auf die ursprüngliche Core Component abgespeichert. Erfolgt eine Veränderung der unterliegenden Core Component durch das Harmonisierungsteam, so ist das ABIE möglicherweise nicht mehr konsistent zu der neuen Definition. Ein weiterer Nachteil von UBL stellt deren Syntaxspezifität dar. *Diese verhindert eine losgelöste Betrachtung der semantischen Struktur von Dokumenten,* wie sie in der CCTS gefordert ist. Zudem verursachen die *Erstellung und Adaption von Dokumenten auf Code-Ebene einen höheren Modellierungsaufwand als für syntaxneutrale Dokumente.* Ein zusätzlicher Nachteil ist in der fehlenden Verankerung von Infrastrukturdiensten in UBL zu sehen: Durch die Tatsache, dass kein einheitliches Repository bereitsteht, in dem Kontextausprägungen und spezifische Varianten abgelegt und vereinheitlicht werden, sind deren Möglichkeiten für Wiederverwendung extrem eingeschränkt. Daraus resultiert ein *hohes Potenzial für inkonsistente Modellierung und redundante Speicherung,* da eine Vereinheitlichung und Überprüfung erstellter Varianten nicht stattfindet, und diese durch den Modellierer individuell gespeichert werden müssen.¹¹² Auch die individuelle Erstellung von Dokumenten ist in UBL problematisch. Anders als in der CCTS, die unter Rückgriff auf die UMM eine *Top-Down*-Identifizierung benötigter ABIE aus Geschäftsprozessen ermöglicht, erfolgt die Dokumentenerstellung in UBL *Bottom-Up*. Kleinere semantische Einheiten werden zu größeren Komponenten zusammengefasst, bis das erwünschte Aggregationsniveau erreicht ist (vgl. Abb. 15 in Kap. 2.4.3). *Durch diese Vorgehensweise werden bedarfsgetriebene Anforderungen an die auszutauschenden Dokumente in geringerem Maße berücksichtigt,* da die Dokumente nicht sukzessive aus einem Ausgangsprozess abgeleitet und verfeinert, sondern umgekehrt aus kleineren Entitäten aggregiert werden. Das resultierende Dokument entspricht somit gegebenenfalls nicht den Anforderungen des Ausgangsprozesses.

Der mit den skizzierten Defiziten einhergehende personelle und monetäre Mehraufwand verringert die Attraktivität von CCTS bzw. UBL besonders für kleine und mittlere Unternehmen, bei denen die für den Erstellungsprozess der Geschäftsdokumente benötigten Ressourcen knapp oder nicht vorhanden sind. Diese Tatsache steht den ursprünglichen Zielen der ebXML-Initiative, den elektronischen Handel unabhängig von der Unternehmensgröße oder -region zu fördern, diametral gegenüber. Einen potenziellen Lösungsansatz für die

¹¹² Dieses Problem existiert auch für die CCTS, bis ein globales Repository verfügbar ist.

Defizite stellt die *konfigurative Referenzmodellierung* dar. Diese beinhaltet ein zentrales Repository für allumfassende Geschäftsdokumentenmodelle und Mechanismen zur konsistenten Generierung von Varianten. Im Folgenden soll daher untersucht werden, ob eine Integration von konfigurativer Referenzmodellierung und CCTS bzw. UBL einen Lösungsweg bietet, um die Defizite bei der Erstellung elektronischer Geschäftsdokumente und ihrer Varianten zu überwinden. Zudem ist zu prüfen, welche weiteren Maßnahmen nötig sind, um die vorkonfigurierten Varianten tatsächlich einzusetzen.

3 Konfiguration elektronischer Geschäftsdokumente mit dem H2-Toolset

3.1 Referenzmodelladaption

Informationsmodelle haben sich in der Vergangenheit als probates Mittel zur Strukturierung unübersichtlicher und komplexer Problembereiche etabliert.¹¹³ Ein Informationsmodell ist das Ergebnis einer Konstruktionsleistung eines Modellierers, der Informationen über zu modellierende Elemente eines Systems zu einer Zeit für Anwendungssystem- und Organisationsgestalter unter Rückgriff auf eine Modellierungssprache als relevant deklariert.¹¹⁴ Der Konstruktionsprozess eines Modells wird als Modellierung bezeichnet.¹¹⁵ In der Wirtschaftsinformatik werden Modelle als Hilfsmittel für die Analyse und Gestaltung von Informationssystemen genutzt.¹¹⁶ Ein *Informationssystem* ist ein System, das den Zwecken der Information und Kommunikation dient.¹¹⁷ Bei der *fachkonzeptionellen Spezifikation* werden diese Systeme in einer formalisierten Sprache modelliert, ohne implementierungstechnische Details oder Restriktionen zu berücksichtigen.¹¹⁸

BECKER und KNACKSTEDT definieren *Referenzmodelle* als Informationsmodelle, die als Ausgangslösungen zur Entwicklung spezifischer Modelle dienen.¹¹⁹ Diese stellen eine spezielle Klasse von Informationsmodellen dar, da sie nicht für einen einzelnen Anwendungskontext erstellt werden, sondern als generalisierte Modelle Informationen für eine Kategorie abstrakter Anwendungsfälle beinhalten.¹²⁰ Somit erheben Referenzmodelle einen Anspruch auf *Allgemeingültigkeit* und *Wiederverwendbarkeit*, da die Konstruktionsergebnisse eines Referenzmodells für einen spezifischen Kontext übernommen und an ihn adaptiert werden können.¹²¹ Daneben beinhalten Referenzmodelle *common-practice*-Wissen über den zugehörigen Anwendungsbereich.¹²² In der Literatur werden einige der Eigenschaften eines Referenzmodells als kritisch bewertet und teilweise zurückgewiesen (wie bspw. das Prinzip der Allgemeingültigkeit¹²³).¹²⁴ Gleichwohl fördert der Einsatz von Referenzmodellen den Transfer betriebswirtschaftlichen Know-Hows, und stellt den Anwendern eine hö-

¹¹³ Vgl. Becker et al. (2002), S. 25.

¹¹⁴ Vgl. Schütte (1998), S. 63.

¹¹⁵ Vgl. Scheer (1997), S. 35.

¹¹⁶ Vgl. Schulze (2001), S. 1 sowie Fettke; Loos (2007), S. 2.

¹¹⁷ Teubner (1999), S. 19.

¹¹⁸ Vgl. Scheer (1992), S. 16 sowie Alpar et al. (2002), S. 270.

¹¹⁹ Vgl. Becker; Knackstedt (2004), S. 39. Weitere Definitionen können bspw. Fettke; Loos (2007), S. 4 entnommen werden.

¹²⁰ Vgl. vom Brocke (2003), S. 34 ff. sowie Becker et al. (2002), S. 25 f.

¹²¹ Vgl. Becker; Niehaves; Knackstedt (2004), S. 1 sowie Fettke; Loos (2007), S. 2.

¹²² Vgl. Becker; Delfmann; Knackstedt (2004a), S. 19.

¹²³ Vgl. vom Brocke; Buddendick (2004b), S. 341.

¹²⁴ Vgl. Fettke; Loos (2007), S. 4.

here Wirtschaftlichkeit der eigenen Informationsmodellierung in Aussicht.¹²⁵ Den erwarteten Einsparungen durch den Einsatz von Referenzmodellen stehen jedoch nicht unerhebliche Adaption- und Wartungskosten gegenüber.¹²⁶ Der notwendige Adaptionaufwand eines Referenzmodells vor dem Einsatz in einem spezifischen Unternehmen steigt im Verhältnis zu seinem Abstraktionsgrad und führt dadurch zu einer geringeren Kundenakzeptanz. Ein spezifischeres Referenzmodell ist jedoch in seinem Adressatenkreis eingeschränkt und beinhaltet daher ein höheres Absatzrisiko.¹²⁷ In der Literatur finden sich verschiedenste Ansätze zur Lösung dieses Referenzmodellierungsdilemmas¹²⁸. Ein vielversprechendes Konzept stellt dabei die konfigurative Referenzmodellierung dar.¹²⁹ Konfigurative Referenzmodelle umfassen das Referenzmodell sowie eine Regelmenge für dessen Adaption. Die Anwendung einzelner Regeln wird dabei durch Konfigurationsparameterausprägungen determiniert, die den individuellen Anwendungskontext des Modells beschreiben. Während des Konfigurationsprozesses wird das Referenzmodell durch die Anwendung der Regeln in ein spezifischeres Modell überführt. Dieser Vorgang entspricht einer Projektion, da die generierten Varianten nur die für den Anwendungskontext relevante Teilmenge des Referenzmodells darstellen.¹³⁰ Konfigurierbare Referenzmodelle können jedoch nicht die Komplexität sämtlicher Anwendungskontexte abdecken. Daher sind situative Faktoren bei der Adaption des Referenzmodells zu berücksichtigen.¹³¹ Diese können eine weitergehende Adaption mit höheren Freiheitsgraden bei der Gestaltung erfordern, wie sie die generische Adaption bietet. Das Referenzmodellierungsdilemma kann nur vermieden werden, wenn sowohl die konfigurative als auch die generische Referenzmodelladaption methodisch unterstützt werden.¹³²

3.1.1 Generische Referenzmodelladaptionstechniken

Bei den generischen Referenzmodelladaptionstechniken existieren beabsichtigte Gestaltungsfreiheiten, die im Rahmen der Variantenerstellung durch den Modellierer ausgefüllt werden.¹³³ Die Mechanismen unterteilen sich in *Aggregation*, *Instanziierung*, *Spezialisierung* und *Analogiekonstruktion*.

¹²⁵ Vgl. Becker; Knackstedt (2003), S. 416 sowie Becker; Knackstedt (2004), S. 39.

¹²⁶ Vgl. Becker; Delfmann; Rieke (2004), S. 2.

¹²⁷ Vgl. Becker et al. (2002), S. 26.

¹²⁸ Vgl. zu diesem Begriff Becker et al. (2002), S. 26.

¹²⁹ Vgl. Delfmann (2006), S. 5.

¹³⁰ Vgl. Delfmann (2006), S. 6.

¹³¹ Vgl. vom Brocke; Buddendick (2004a), S. 25

¹³² Vgl. Becker; Delfmann; Knackstedt (2004b), S. 251.

¹³³ Vgl. Becker; Delfmann; Knackstedt (2004b), S. 252.

Aggregation

Dem Mechanismus der Aggregation liegt die Annahme zugrunde, dass ein integriertes Gesamtmodell aus Modellbausteinen zusammengefügt ist.¹³⁴ Im Rahmen des Adaptionprozesses werden diese vom Modellierer in Abhängigkeit eines konkreten Anwendungskontexts über definierte Schnittstellen zu neuen Lösungen zusammengefügt.¹³⁵

Instanziierung

Instanziierbare Referenzmodelle beinhalten als abstrakt definierte Informationsobjekte, die im Zuge des Adaptionmechanismus gefüllt werden.¹³⁶ Der Referenzmodell Anwender ist dabei auf instanziiere Bereiche innerhalb des Referenzmodells hinzuweisen, die durch ihn zu konkretisieren sind.¹³⁷

Spezialisierung und Analogiekonstruktion

Die Spezialisierung ermöglicht die inhaltlich freie Erweiterung oder Verfeinerung von bewusst vage formulierten Modellelementen.¹³⁸ Bei dem Analogieschluss wird eine kreative Lösung für einen neu zu modellierenden Bereich auf Basis von bereits bestehenden Modellteilen, die inhaltliche oder strukturelle Ähnlichkeiten aufweisen, konstruiert. Dabei werden einzelne Eigenschaften des Modellteils wiederverwendet.¹³⁹ Die Spezialisierung und die Analogiekonstruktion besitzen die größten Modellierungsfreiheiten innerhalb des Adaptionprozesses von Referenzmodellen.¹⁴⁰ Spezifische Regeln bzgl. der Veränderung eines Modells fehlen weitgehend. Mit steigender Modellierungsfreiheit wächst jedoch die Gefahr inkonsistenter Modellierung, die durch entsprechende Konsistenzsicherungsmechanismen zu vermeiden ist.

3.1.2 Konfigurative Referenzmodelladaptionstechniken

Die konfigurativen Referenzmodelladaptionstechniken (im Folgenden unter dem Begriff „konfigurative Referenzmodellierung“ zusammengefasst) ermöglichen die Erstellung spezifischer Modellvarianten in Abhängigkeit von Konfigurationsparameterausprägungen durch Projektion eines integrierten Gesamtmodells.¹⁴¹ Unter Konfigurationsparametern

¹³⁴ Vgl. Becker; Delfmann; Knackstedt (2004b), S. 258 sowie Delfmann (2006), S. 166.

¹³⁵ Vgl. Knackstedt (2006), S. 272.

¹³⁶ Vgl. Becker; Delfmann; Knackstedt (2004a), S. 20.

¹³⁷ Vgl. Becker; Delfmann; Knackstedt (2004b), S. 258.

¹³⁸ Vgl. Becker; Delfmann; Knackstedt (2004a), S. 20.

¹³⁹ Vgl. Becker; Delfmann; Knackstedt (2004a), S. 20.

¹⁴⁰ Vgl. Becker; Delfmann; Knackstedt (2004b), S. 259.

¹⁴¹ Vgl. Becker; Delfmann; Knackstedt (2004b), S. 254.

werden dabei jene Eigenschaften eines Anwendungskontexts verstanden, anhand derer ein Referenzmodell konfiguriert werden kann. Diese unterteilen sich in Unternehmensmerkmale und Perspektiven. *Unternehmensmerkmale* und ihre Ausprägungen repräsentieren Klassen von Unternehmen, die durch das Referenzmodell repräsentiert werden.¹⁴² Im Rahmen der Konfiguration ordnet der Modellanwender das von ihm untersuchte Unternehmen hinsichtlich der vorhandenen Unternehmensmerkmale und deren Ausprägungen innerhalb des Referenzmodells ein. Neben den Klassen von Unternehmen sind die agierenden Nutzer und ihre spezifischen Sichten auf das Modell innerhalb dieser zu identifizieren. Diese *Perspektiven* repräsentieren verschiedene Sichtweisen auf das Referenzmodell, die sich durch unterschiedliche Verwendungskontexte ergeben.¹⁴³ Bedeutende, zueinander orthogonale Kategorien für die Perspektivenbildung sind *Rollen* wie Fachkonzeptersteller oder Implementierer, *Zwecke* wie Organisations- oder Anwendungssystemgestaltung und *sonstige Einflüsse* wie Rot-Grün-Blindheit.¹⁴⁴

Um ein Referenzmodell effizient konfigurieren zu können, ist eine Unterstützung von Mechanismen mit unterschiedlich breitem Wirkungsgrad erforderlich.¹⁴⁵ Diese Konfigurationsmechanismen unterteilen sich in die Modelltypselektion, Elementtypselektion, Elementselektion, Bezeichnungsvariation und Darstellungsvariation.¹⁴⁶

Modelltypselektion

Bei der Modelltypselektion erfolgt eine perspektivenspezifische Ausblendung irrelevanter Modelltypen.¹⁴⁷ Modelltypen repräsentieren Arten von Modellierungstechniken, die in einem Referenzmodell kombiniert werden können, wie z. B. Fachbegriffsmodelle, Organigramme, *Entity-Relationship-Modelle (ERM)* oder ereignisgesteuerte Prozessketten.¹⁴⁸

Elementtypselektion

Die Elementtypselektion ermöglicht die Ausblendung von Elementtypen auf der Metamodellebene.¹⁴⁹ Dadurch ist es möglich, Varianten zu einer Modellierungssprache durch Eliminierung irrelevanter Elementtypen zu generieren. Als Grundlage für diesen Mechanismus dienen Standard-Modelltypvarianten, deren Elementtypen abhängig von der Perspek-

¹⁴² Vgl. Becker et al. (2002), S. 27.

¹⁴³ Vgl. Becker et al. (2002), S. 28.

¹⁴⁴ Vgl. Knackstedt (2006), S. 215 f.

¹⁴⁵ Vgl. Becker; Delfmann; Knackstedt (2004b), S. 254.

¹⁴⁶ Fachkonzeptionelle Details dieser Mechanismen sind Anhang C zu entnehmen.

¹⁴⁷ Vgl. Delfmann (2006), S. 97.

¹⁴⁸ Vgl. Becker; Delfmann; Knackstedt (2004b), S. 254.

¹⁴⁹ Vgl. Becker; Delfmann; Knackstedt (2004b), S. 254.

tive beschränkt werden.¹⁵⁰ Die Ausdrucksmächtigkeit einer Sprache wird somit eingeschränkt.

Elementselektion

Bei der Elementselektion werden im Gegensatz zu der Modelltyp- und Elementtypselektion nicht Elementtypen, sondern deren Instanzen ausgeblendet.¹⁵¹ Der Mechanismus operiert somit auf der Modellebene und unterteilt sich in vier verschiedene Selektionskriterien.¹⁵² Die *Elementselektion über Typen* berücksichtigt die Tatsache, dass sowohl Elemente als auch Beziehungen innerhalb eines Referenzmodells bestimmten Typen angehören.¹⁵³ Die Typisierung von Elementen kann auf zwei Arten erfolgen. Einerseits kann diese durch eine konkrete Modellierungssprache bereits vorgegeben sein, andererseits ist eine manuelle Typisierung durch den Modellierer mittels der Spezifikation individueller Beziehungen zwischen den Elementen möglich. Bei diesem Konfigurationsmechanismus steht letzte Variante im Fokus, wobei die Existenz der Beziehungen an eine spezifische Perspektive gebunden ist. Bei der *Elementselektion mittels Hierarchiestufenaggregation* werden differenzierte Detaillierungsgrade innerhalb hierarchischer Strukturen von Modellelementen in Abhängigkeit zu einer Perspektive generiert. Dabei wird für jede Perspektive die maximale Hierarchiestufe eines Elementsystems festgelegt, um im Anwendungsfall des Konfigurationsmechanismus tiefer in der Hierarchie liegende Elemente auf diese Stufe zu aggregieren. Bei der *Elementselektion über Attribute* erfolgt eine Selektion von Elementen anhand direkt zugeordneter Eigenschaften. Die *Elementselektion nach Konfigurationstermen* schließlich ermöglicht die Zuordnung von Konfigurationsparametern zu Modellelementen. Die Zuordnung erfolgt dabei indirekt über einen Konfigurationsterm, der eine logische Verknüpfung von Konfigurationsparametern darstellt. Für die Erstellung des Terms wird auf eine vordefinierte kontextfreie Grammatik zurückgegriffen, wie bspw. die folgende, an SCHWEGMANN angelehnte Formalisierung (vgl. Abb. 18):

```

<Term> ::= (<Ausdruck>{<Operator><Term>});
<Ausdruck> ::= {<Präfix><Konfigurationsparameter><KP-Ausprägungsliste>;
<KP-Ausprägungsliste> ::= „(“ <Präfix> <KP-Ausprägungsliste> {<Operator> <Präfix> <KP-Ausprägungsliste>} „)“;
<KP-Ausprägungsliste> ::= <KP-Ausprägung>;
<Präfix> ::= „NOT“;
<Operator> ::= „AND“ | „OR“ | „XOR“;
    
```

Abb. 18: Kontextfreie Grammatik zur Formulierung von Konfigurationstermen¹⁵⁴

¹⁵⁰ Vgl. Delfmann (2006), S. 100.

¹⁵¹ Vgl. Knackstedt (2006), S. 219.

¹⁵² Vgl. im Folgenden Becker et al. (2002), S. 73 f.

¹⁵³ Vgl. Becker et al. (2002), S. 98.

¹⁵⁴ In Anlehnung an Schwegmann (1999), S. 144 ff.

Bezeichnungsvariation

Die Bezeichnungsvariation ermöglicht die Normierung der verwendeten Begriffe innerhalb eines Referenzmodells.¹⁵⁵ Dazu gehört in einem weiteren Sinne auch die kontrollierte Einführung von Synonymen.¹⁵⁶ Der Konfigurationsmechanismus ermöglicht dabei den konfigurationsparameterspezifischen Austausch äquivalenter Bezeichnungen für ein Objekt.¹⁵⁷ Dadurch können in der Praxis synonym verwendete Begriffe wie bspw. Rechnung und Faktura konfigurationsparameterspezifisch ausgetauscht werden.¹⁵⁸ Die zu verwendende Bezeichnung innerhalb einer Synonymgruppe, die mehrere Synonymbezeichnungen zusammenfasst, wird dabei erkannt und innerhalb des Modells eingesetzt.

Darstellungsvariation

Im Gegensatz zu den vorangegangenen Konfigurationsmechanismen modifiziert die Darstellungsvariation nicht die konzeptionellen, sondern die visuellen Aspekte einer Sprache in Abhängigkeit einer spezifischen Perspektive. Es werden drei Arten von Darstellungsvariationen unterschieden. Bei der *Darstellungsvariation der Symbole* werden demselben Sprachaspekt unterschiedliche Symbole zugeordnet. Dabei können sowohl die Darstellungen kompletter Sprachen als auch nur einzelner Objekttypen ausgetauscht werden. Der Mechanismus der *Darstellungsvariation der Topologie* ermöglicht die Variation in der topologischen Anordnung von Modellen, wie bspw. die von der Standardnotation der erweiterten EPK abweichende Spaltendarstellung. Der dritte Aspekt umfasst die *Darstellungsvariation der Konfigurationsansatzpunkte*, deren Repräsentation sich ebenfalls unterschiedlich darstellen lässt, z. B. durch Build-Time-Operatoren oder in Spaltendarstellung.

Die Erstellung und Konfiguration von Referenzmodellen erweisen sich als sehr komplex, so dass es einer durchgehenden Unterstützung durch computergestützte Werkzeuge bedarf. Ein Metamodellierungswerkzeug mit konfigurativer Komponente stellt das H2-Toolset dar, das Gegenstand des folgenden Kapitels ist.

3.1.3 Konfigurative Referenzmodellierung mit dem H2-Toolset

Das H2-Toolset ist ein am Lehrstuhl für Wirtschaftsinformatik und Informationsmanagement der Westfälischen Wilhelms-Universität entwickeltes, softwarebasiertes Metamodellierungswerkzeug, mit dem hierarchische Modellierungssprachen konstruiert und diesen

¹⁵⁵ Vgl. Knackstedt (2006), S. 260

¹⁵⁶ Knackstedt (2006), S. 260.

¹⁵⁷ Vgl. Becker; Delfmann; Knackstedt (2004b), S. 257.

¹⁵⁸ Vgl. Becker et al. (2002), S. 74.

Sprachdefinitionen entsprechende Modelle erstellt werden können.¹⁵⁹ Die Hauptkomponenten des H2-Toolsets umfassen einen *Spracheditor* für die Erstellung der Modellierungssprache sowie einen *Modelleditor* für die zugehörigen Modelle. Das entsprechende ERM mit erweiterter Min-Max-Notation des Kernrepositorys kann Abb. 19 entnommen werden.¹⁶⁰

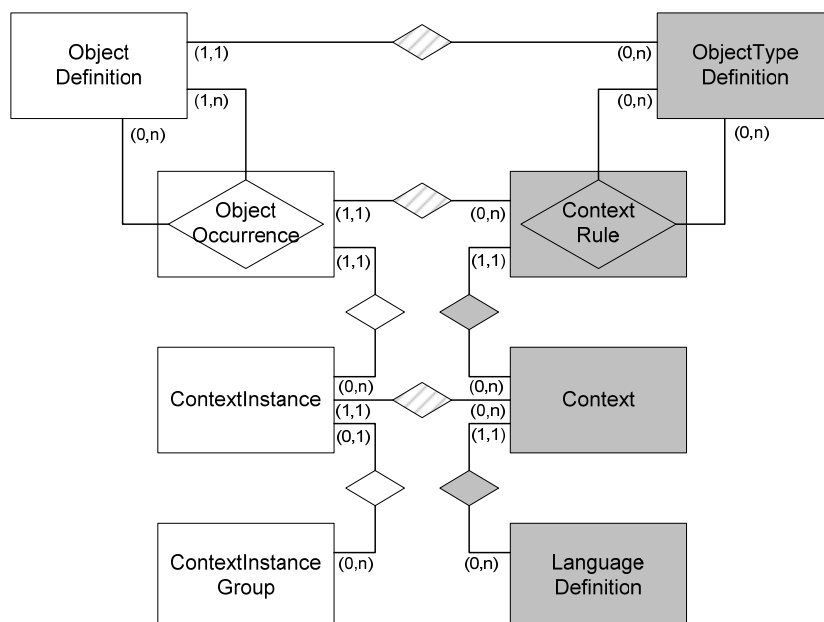


Abb. 19: Hauptkomponenten des H2-Kernrepositorys¹⁶¹

Der rechte Teil des H2-Kernrepositorys enthält die fachkonzeptionelle Beschreibung von Modellierungssprachen, und kann daher als sprachbasiertes Meta-Metamodell aufgefasst werden.¹⁶² Eine Sprache umfasst eine Menge von Elementen, die Objekttypdefinitionen (Entitytyp (*ET*) *ObjectTypeDefinition*).¹⁶³ Zwischen diesen Objekttypdefinitionen existieren Beziehungen, die durch Kontextregeln (*ET Context Rule*) repräsentiert werden. Eine Kontextregel enthält genau eine Vorgänger-Nachfolger-Beziehung zwischen zwei Elementen und beinhaltet mehrere Möglichkeiten, um die Existenz eines Elements zu charakterisieren: Die Regel *CreateDefinition* erlaubt die Erstellung der Elementdefinition innerhalb des aktuellen Kontexts. Eine *CreateOccurrence*-Regel ermöglicht dem Modellierer eine bereits definierte Modellinstanz durch die Erstellung einer Ausprägung in einem unterschiedlichen Kontext wiederzuverwenden. Die Verwendung einer *CreateOccurrenceExplicit*-Regel erstellt ebenfalls eine Ausprägung, deren Komponenten weiter spezialisiert werden können. Dazu wird die *Occurrence Explicit* bei ihrer Erstellung von der Definition getrennt.

¹⁵⁹ Vgl. Becker et al. (2005), S. 10.

¹⁶⁰ Zu Entity-Relationship-Modellen vgl. Chen (1976).

¹⁶¹ Quelle: Becker et al. (2006b), S. 8.

¹⁶² Vgl. Becker et al. (2006b), S. 7.

¹⁶³ Vgl. Becker et al. (2006b), S. 7 f.

Kontextregeln können zu einem Block zusammengefasst werden, dem Kontext (ET *Context*). Dieser übernimmt sowohl Filter- als auch Containerfunktionen. Als Container beinhaltet er die modellierten Elemente und deren Beziehungen. In seiner Funktion als Filter schränkt er mögliche Beziehungen zwischen Elementen innerhalb konkreter Instanzen des Kontexts ein.¹⁶⁴ Hinsichtlich ihrer Eigenschaften können vier Kontextarten unterschieden werden:¹⁶⁵

- *Grundlegende Kontexte* definieren Elemente, die in anderen Kontexten wiederverwendet werden, ohne für die Definition Komponenten anderer Kontexte zu referenzieren.
- *Verbindende Kontexte* weisen multiple Relationen auf, d. h. sie verwenden Elemente aus anderen Kontexten wieder und werden parallel wiederverwendet.
- *Aggregierende Kontexte* fügen Komponenten aus anderen Kontexten auf der höchsten Hierarchieebene zu einer zusammengehörigen Komponente zusammen.
- *Einzelne Kontexte* weisen keinerlei Beziehungen zu anderen Kontexten auf.

Mehrere Kontexte werden zu einer Modellierungssprache zusammengefasst (ET *Language Definition*). Neben den Elementen und ihren Beziehungen können ersteren im Spracheditor zusätzliche Attribute und Attributgruppen sowie Kantentypen zugewiesen werden, um sie semantisch zu erweitern.

Konkrete Modelle einer Modellierungssprache fungieren als Metamodelle zur Realwelt. Ist die Modellierungssprache spezifiziert, können im Modelleditor dieser Sprachdefinition entsprechende Modelle erzeugt werden (linker Teil des H2-Kernrepositories). Dies geschieht durch die Erzeugung von Instanzen der Objekttypdefinitionen, den Objektdefinitionen (ET *ObjectDefinition*). Die Beziehung zwischen zwei Objektdefinitionen wird durch den uminterpretierten ET *ObjectOccurrence* konkretisiert. Diese basieren auf einer Kontextregel und sind Teil einer Kontextinstanz (ET *ContextInstance*). Kontextinstanzen können wiederum zu Kontextinstanzgruppen (ET *ContextInstanceGroup*) zusammengefasst werden. Diesen Gruppen können Regeln annotiert werden, wie bspw. die Forderung nach identischen Blattmengen.¹⁶⁶ Eine weitere Funktion des H2-Toolsets stellt der Hierarchieeditor dar, mit dem Hierarchiesets und -ebenen definiert werden können. Zudem ist es möglich, Sprach- und Modelldefinitionen im Toolset flexibel über eine XML-Schnittstelle

¹⁶⁴ Vgl. Becker; Janiesch; Pfeiffer (2007), S. 3.

¹⁶⁵ Vgl. Becker; Janiesch; Pfeiffer (2007), S. 3.

¹⁶⁶ Vgl. Becker et al. (2006a), S. 11.

zu im- und exportieren.¹⁶⁷ Ein konkretes Beispiel für eine Sprache und ein Modell im H2-Toolset ist Abb. 20 zu entnehmen.

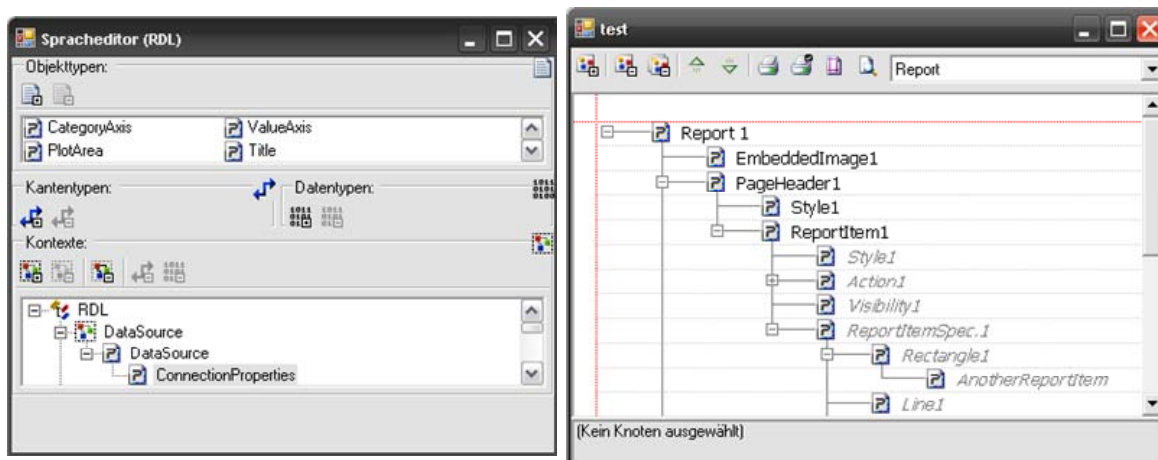


Abb. 20: Sprach- und Modelleditor im H2-Toolset

Aufbauend auf die im vorangegangenen Kapitel vorgestellten Grundfunktionalitäten wurde das H2-Toolset um die in Kap. 3.1.2 vorgestellten Methoden der konfigurativen Referenzmodellierung erweitert.¹⁶⁸ Zusätzlich zu den Mechanismen Modelltypselektion, Elementtypselektion, Elementselektion, Bezeichnungsvariation und Darstellungsvariation wurde eine Modellselektion implementiert, die konkrete Instanzen einer Modellierungssprache ausblendet. Die zugehörige Sprache und andere in dieser Sprache erstellte Modelle bleiben von der Modellselektion unberührt.¹⁶⁹

Mit dem H2-Toolset lassen sich hierarchische Modellierungssprachen und zugehörige Modelle erstellen. Geschäftsdokumente nach CCTS bzw. UBL besitzen einen hierarchischen Aufbau und sind daher für eine Umsetzung im H2-Toolset geeignet. Um die Integrationsmöglichkeiten zwischen konfigurativer Referenzmodellierung und der CCTS bzw. UBL zu untersuchen, sind diese zunächst in Sprachen und Modelle des Toolsets zu überführen.

3.2 Sprachspezifikation im H2-Toolset

3.2.1 Modellierung der CCTS

Das ERM in Abb. 21 spezifiziert die ursprüngliche CCTS-Sprache fachkonzeptionell.¹⁷⁰

¹⁶⁷ Vgl. Becker et al. (2007) (Im Erscheinen).

¹⁶⁸ Vgl. Becker et al. (2006a), S. 4.

¹⁶⁹ Eine ausführliche Darstellung der Konfigurationsmechanismen im H2-Toolset ist Becker et al. (2006a) zu entnehmen.

¹⁷⁰ Von unbedeutenden Trivialattributen wurde in der Darstellung abstrahiert.

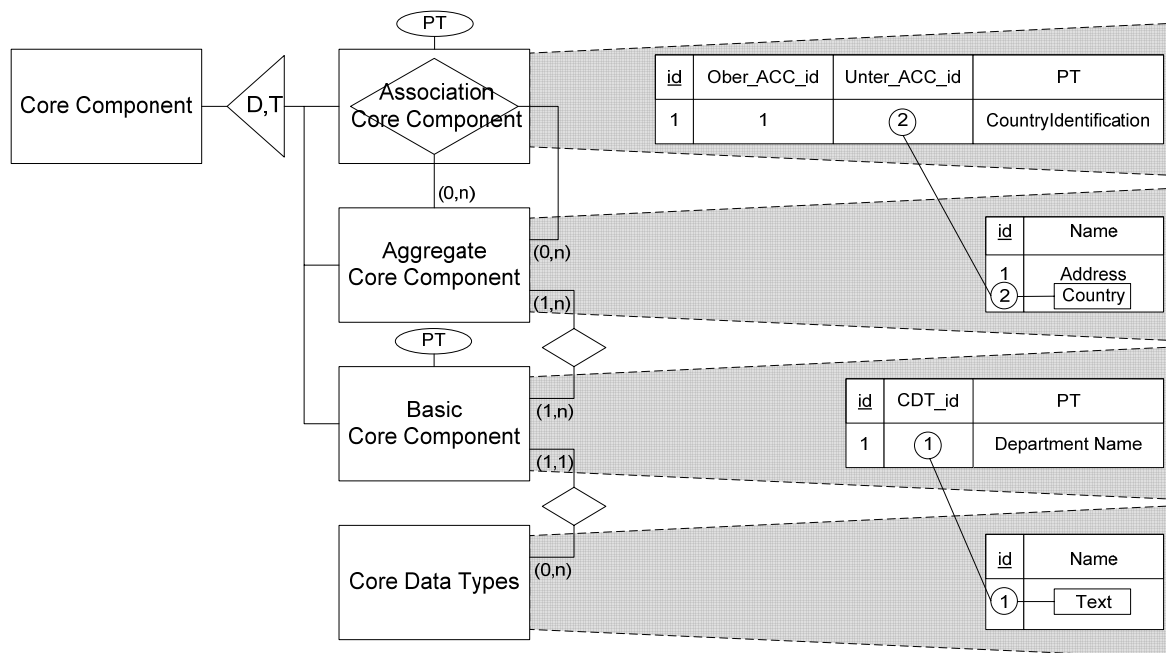


Abb. 22: Identifikation des impliziten Representation Terms

Die rechte, eingefärbte Seite des ERM repräsentiert das Konzept kontextspezifischer Business Information Entities (vgl. wiederum Abb. 21). Die Struktur der ABIE, ASBIE und BBIE ist mit Ausnahme des zusätzlich annotierten Qualifier Terms (Attribut *QT*) identisch zu ihren kontextneutralen Analogons. Die gestreiften Relationen spezifizieren die Beziehungen zwischen der kontextneutralen und der kontextspezifischen Domäne. ABIE (ET *Aggregate Business Information Entity*), BBIE (ET *Basic Business Information Entity*) und BDTs (ET *Business Data Type*) sind aus genau einer ACC, BCC bzw. einem CDT abgeleitet. Umgekehrt kann eine ACC, BCC oder ein CDT jedoch als Basis für beliebig viele kontextspezifische Varianten dienen.

Der ET *Context Category* beinhaltet die Kontextkategorien nach CCTS. Um die hierarchischen Beziehungen innerhalb einzelner Kategorien abzubilden, wurde eine entsprechende Hierarchie modelliert. Der ET *Context Value* spezifiziert entsprechende Kontextausprägungen. Der uninterpretierte Relationshiptyp *CC-CV-Rel.* enthält die für Variantengenerierung relevante Kombinationen von Kontextkategorien und -ausprägungen. Kontextkategorien können beliebig viele Kontextausprägungen enthalten. Aufgrund der disjunkten Struktur der Kategorien sollten Ausprägungen jedoch im Normalfall nur genau einer Kategorie angehören. Wenn Werte keiner Kategorie zugeordnet werden können, so sollten sie in einer gesonderten Kategorie (z. B. *others*) zusammengefasst werden. Da Kombinationen von Kontextkategorien und -ausprägungen für alle Arten von BIE relevant sind, wird eine entsprechende Relation an die Generalisierung *Business Information Entity* modelliert. Auch Business Data Types stehen mit der *CC-CV-Rel.* in Beziehung.

Um das im H2-Toolset realisierte Prinzip der konfigurativen Referenzmodellierung mit dem CCTS-Konzept zu integrieren, ist das ERM der CCTS entsprechend zu adaptieren (vgl. Abb. 23).

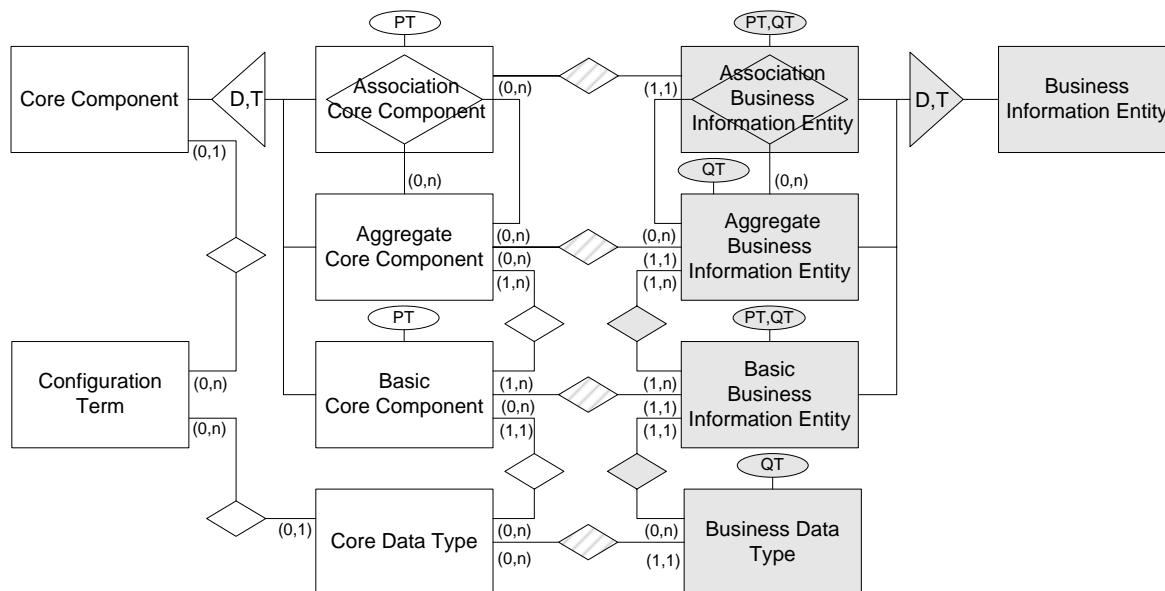


Abb. 23: ERM des adaptierten CCTS-Konzepts für das H2-Toolset

Die kontextspezifische Domäne wird im H2-Toolset nicht separat modelliert, sondern resultiert aus der Konfiguration der kontextneutralen Komponenten. Zudem wird ein Konfigurationsterm (ET *Configuration Term*) als Analogon zu der logischen Verknüpfung von Konfigurationskategorien und Ausprägungen direkt an die Core Component bzw. den Core Data Type annotiert. Die Kardinalitäten ändern sich entsprechend den Restriktionen des H2-Toolsets, dass ein Element maximal mit einem Term belegt werden darf. Die Unterschiede der ERMs visualisieren die Divergenz zwischen dem konventionellen CCTS-Ansatz und der konfigurativen Referenzmodellierung. Während ersterer durch eine inverse Beziehung zwischen Business Information Entities, Kontextausprägungen und zugehörigen Core Components gekennzeichnet ist, erfolgt bei letzterer die Erstellung von Business Information Entities direkt durch die Annotation von Termen an die Core Components.

Für die Modellierung im H2-Toolset ist zu klären, welche Komponenten explizit als Sprachtyp modelliert, und welche implizit durch die Struktur, Attribute, Konfigurationsmechanismen etc. repräsentiert werden. Folgende Annahmen wurden getroffen:

- Aggregate Core Components, Association Core Components (Aggregation und Komposition) und Basic Core Components werden als eigenständige Sprachkonstrukte modelliert. Durch die Verwendung dieser CCTS-Kernkomponenten als Sprachelemente können Dokumentenmodelle erstellt werden, die konform zu den formulierten Regeln der CCTS sind.

- Bei Property- und Representation Terms ist zwischen impliziter und expliziter Modellierung abzuwägen. Da der Representation Term immer eindeutig durch die zugehörige ACC (bei ASCC) oder den CDT (Bei BCC) determiniert ist, wird er analog zu dem CCTS-ERM nicht als eigener Objekttyp modelliert. Property Terms können während des Konfigurationsprozesses qualifiziert werden. Dieser Vorgang wäre bei impliziter Modellierung durch Annotation von Attributen nicht realisierbar. Die explizite Modellierung als Objekttyp in Kombination mit Konfigurationsmechanismen erscheint daher sinnvoll.
- Qualifier für Object Classes und Representation Terms treten nur auf kontextspezifischer Ebene bei Basic- sowie Association Business Information Entities und Business Data Types auf, und präzisieren dort ihre semantische Bedeutung. Qualifier Terms decken aber nicht den gesamten Definitionsraum der CCTS ab, da sie innerhalb von Core Components nicht auftreten. Ihre Realisierung erfolgt daher implizit über den Konfigurationsmechanismus der Bezeichnungsvariation, mit dem die Angaben innerhalb der OCs und PTs kontextspezifisch modifiziert werden (vgl. Kap. 3.5).
- Core Data Types und ihre Content- und Supplementary Components besitzen eine große Bedeutung innerhalb der CCTS, da alle Core Components letztlich auf Core Data Types basieren. Zudem können Content- und Supplementary Components bei der Transformation in Business Data Types beschränkt werden. Diese beiden Tatsachen legen die Entscheidung nahe, CDTs inklusive Content und Supplementary Components als eigene Objekttypen zu modellieren.
- Content- und Supplementary Components eines Core Data Types werden technisch durch primitive Datentypen repräsentiert. Die Modellierung letzterer erfüllt eine wichtige Informationsfunktion für den Modellierer und wurde explizit als eigener Objekttyp gewählt, auch wenn eine implizitere Modellierung als Attribut denkbar wäre. Die Restriktionen, die bei einer Transformation zu einem Business Data Type spezifiziert werden können, werden als Attribut *Restriction/Restriction Value* modelliert. Dieses Attribut enthält eine Liste von Tupeln aus der Restriktion und dem entsprechenden Wert. Restriktionen werden nicht als explizite Objekttypen modelliert, da sie in der kontextneutralen Domäne nicht auftreten, und der Objekttyp dort außer einer Platzhalterfunktion keine Bedeutung besäße. Die Modellierung als Attribut hat jedoch den Nachteil, dass eine automatische Konfiguration dieser Restriktionen ausgeschlossen ist.
- Da das H2-Toolset nach aktuellem Entwicklungsstand keine explizite Modellierung von Kardinalitäten unterstützt, wurden Hilfskonstrukte in Form von Kantentypen

definiert, die den Kardinalitäten (0:1), (0:n), (1:1), (1:n) entsprechen. Diese Kanten-
typen werden im Spracheditor den Vorgänger-Nachfolger-Beziehungen annotiert.¹⁷¹

- Auf Modellebene kann der Modellierer den Modellelementen Kantentypen entsprechend den im Spracheditor formulierten Möglichkeiten zuweisen. Dieses Hilfskonstrukt erfüllt jedoch nur eine visualisierende Funktion, konkrete Restriktionen bei der Modellierung von Elementen auf Modellebene werden nicht unterstützt. So ist es z. B. nicht möglich, die Content Component eines Core Data Types als Pflichtelement zu kennzeichnen. Zwar kann ihr auf Sprachebene der Kantentyp (1:1) zugewiesen werden, doch ist zu berücksichtigen, dass auf Modellebene standardmäßig keine Kantenbeziehung spezifiziert ist und diese erst optional zuzuweisen sind. Doch auch durch diese Zuweisung wird die Kardinalität nur dargestellt. Eine ontologische Prüfung, wie sie bspw. bei einem Löschvorgang von Pflichtfeldern denkbar wäre, findet nicht statt. Abb. 24 verdeutlicht diesen Sachverhalt: Obwohl die Content Component Identifier.Content als Pflichtfeld markiert ist (1:1), wäre es möglich, diese zu löschen.

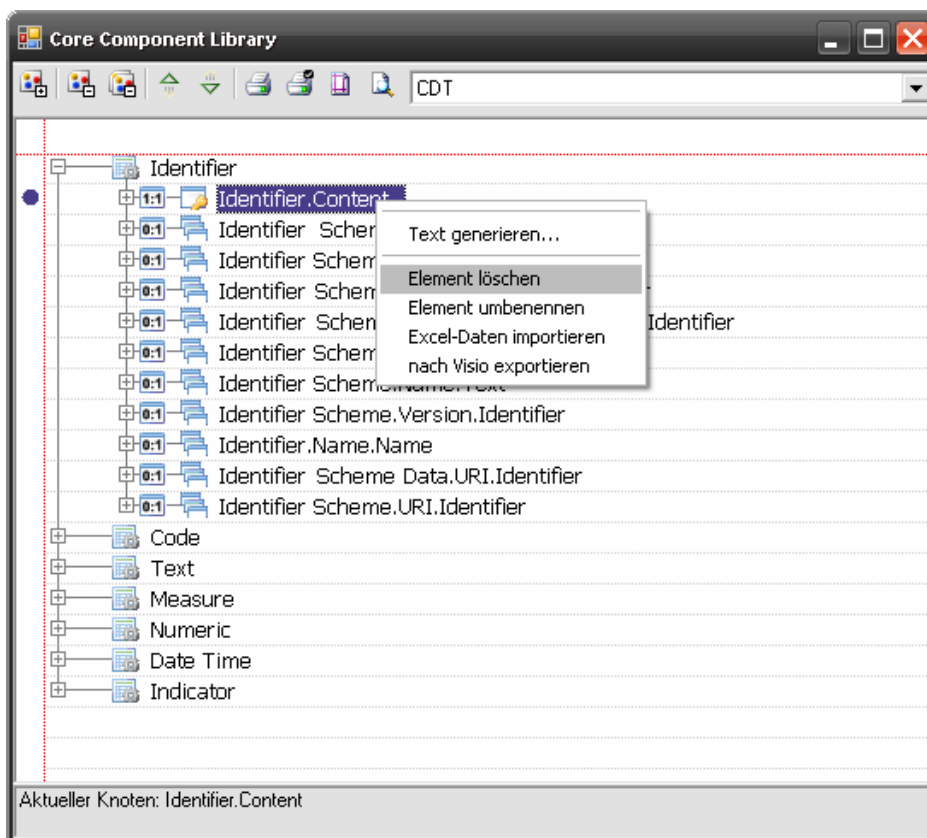


Abb. 24: Visualisierender Aspekt der Kardinalitäten im Modelleditor

¹⁷¹ Für eine verfeinerte Darstellung der Kardinalitäten könnten bei Bedarf weitere Kantentypen modelliert werden.

Im zweiten Schritt der Spracherstellung sind die Kontextstrukturen der CCTS-Sprache zu definieren und die Elemente innerhalb dieser miteinander in Beziehung zu setzen.¹⁷² Dabei ist zwischen Ausdrucksstärke, Wiederverwendbarkeit und Übersichtlichkeit der Sprachdefinition sowie der resultierenden Modelle abzuwägen. Folgende Struktur wurde modelliert (vgl. im Folgenden Abb. 25):¹⁷³

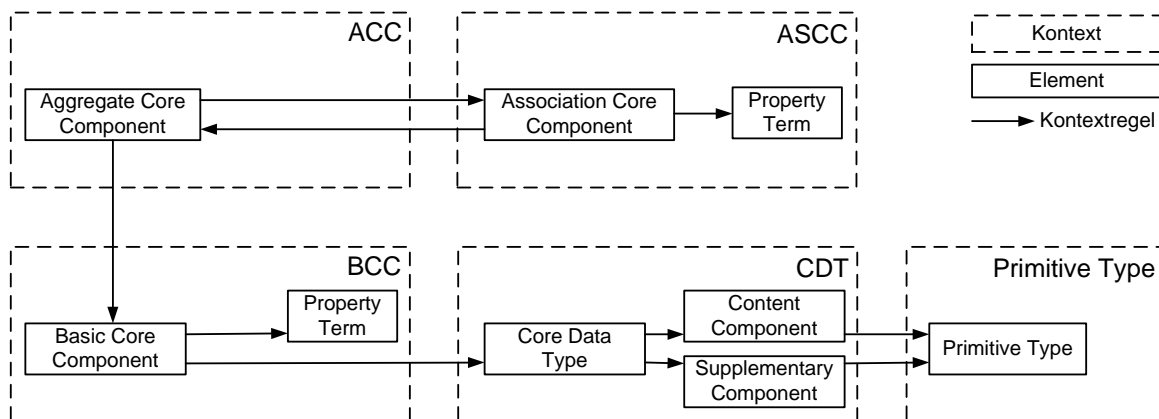


Abb. 25: Kontext- und Elementstruktur der CCTS

- Der *grundlegende Kontext* innerhalb der CCTS-Sprache lautet *Primitive Type*. Da jede Komponente innerhalb der CCTS letztlich aus primitiven Datentypen besteht, und während der Sprachtypmodellierung die Entscheidung getroffen wurde, primitive Datentypen explizit zu modellieren, werden diese innerhalb des gleichnamigen Kontexts definiert.
- Innerhalb des Kontexts *CDT* werden Core Data Types sowie Content- und Supplementary Components definiert. Letztere sind genau einem Core Data Type zugeordnet, ihre Wiederverwendungsmöglichkeiten daher gering. Deshalb besitzen sie auch keinen eigenen Kontext. Bei der offiziellen Spezifikation der Core Data Types weicht der Object Class Name innerhalb der Supplementary Components in einigen Fällen von der Oberklasse ab (z. B. bei der SC *Identifier Scheme* des CDTs *Identifier*). Daher wäre abzuwägen, die Object Class für Supplementary Components ergänzend zu modellieren. Aus Gründen der Übersichtlichkeit wird diese Alternative jedoch verworfen. Stattdessen werden Content- und Supplementary Components durch das Schema `Object Class.Property Term.Representation Term` inhaltlich beschrieben (vgl. Abb. 24). Die Beziehung zwischen Content- und Supplementary Components und ihren Datentypen

¹⁷² Diese Kontexte sind nicht mit den Kontextkategorien zu verwechseln, die bei der Variantengenerierung der CCTS und UBL verwendet werden.

¹⁷³ Die Kontexte werden in den folgenden Aufzählungen kursiv dargestellt, um sie von gleichnamigen Objekttypen zu unterscheiden.

wird durch die Zuweisung von Occurrences des Objekttyps Primitive Type dargestellt. Da *CDT* sowohl Kontexte wiederverwendet als auch wiederverwendet wird, stellt er einen *verbindenden Kontext* dar.

- Ebenfalls *verbindende Funktion besitzt der Kontext BCC*. Er beinhaltet die Definition des Objekttypen Basic Core Component sowie des zugehörigen Property Terms. Property Terms erhalten keinen eigenen Kontext, da sie spezifisch für eine BCC gelten und daher keinen hohen Wiederverwendungswert besitzen. Zudem werden CDT-Occurrences ohne expliziten Representation Term unter die Basic Core Component modelliert.
- Die Kontexte *ASCC (Aggregation)* und *ASCC (Composition)* sind prinzipiell identisch zu dem Kontext *BCC* modelliert. Statt CDT-Occurrences werden den *ASCC* jedoch *ACC* als implizite Representation Terms zugewiesen. Alternativ wären *ASCC* auch als *ACC* mit Attributen modellierbar, die als Ausprägungen die Werte *Aggregation* oder *Composition* erhielten. Die bei der Konfiguration notwendige Unterscheidung, Aggregationen in ihrer ursprünglichen Rolle als *ACC* nicht auszublenzen, wäre so jedoch nicht realisierbar. Zudem besitzen originäre *ACC* keinen Property- und Representation Term, die für *ASCC* implizit über Attribute zu modellieren wären. Da die aufgezeigten Defizite einer konsistenten Modellierung entgegen stehen, wird diese Alternative verworfen. Der *Kontext ASCC wäre zwar ein verbindender Kontext*, passt jedoch nicht optimal in das Klassifikationsschema (vgl. hierzu Kontext *ACC*).
- Der Kontext *ACC* definiert den Objekttyp *ACC*, der die Komponenten aus den Kontexten *BCC*, *ASCC (Composition)* und *ASCC (Aggregation)* zu semantischen Strukturen zusammenfasst. *ACC* wäre eigentlich ein *verbindender Kontext*, da *ACC* in dem Kontext *ASCC* wiederverwendet werden. Zwischen den Kontexten *ACC* und *ASCC* existieren jedoch wechselseitige Referenzierungen, da *ASCC* *ACC*-Occurrences beinhalten und vice versa. Abstrakter betrachtet referenzieren *ACC* letztlich auf *ACC*, da *ASCC* nur Hilfskonstrukte darstellen, die mit zusätzlichen Informationen angereichert werden. Ein solcher Kontext könnte daher als *rekursiv aggregierend* klassifiziert werden. Durch diese Eigenschaft ist es möglich Endlosschleifen in dem Kontext zu produzieren (vgl. Abb. 26). Diese können entweder erwünscht (wie bei einigen Definitionen der *UBL-Common-Library*) oder unerwünscht sein.

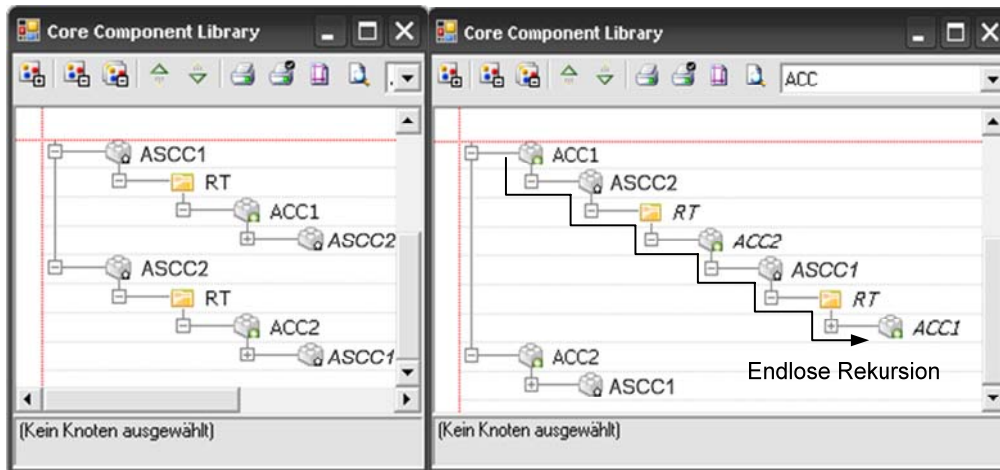


Abb. 26: Endlosschleife durch gegenseitige Referenzierung

Ein Überblick über die Objekttypen, Kantentypen, Beziehungen und Kontexte der modellierten CCTS-Sprache im H2-Toolset ist Abb. 27 zu entnehmen. Um Gemeinsamkeiten und Unterschiede zwischen CCTS und UBL zu verdeutlichen, folgt die Spezifikation letzterer im folgenden Kapitel.

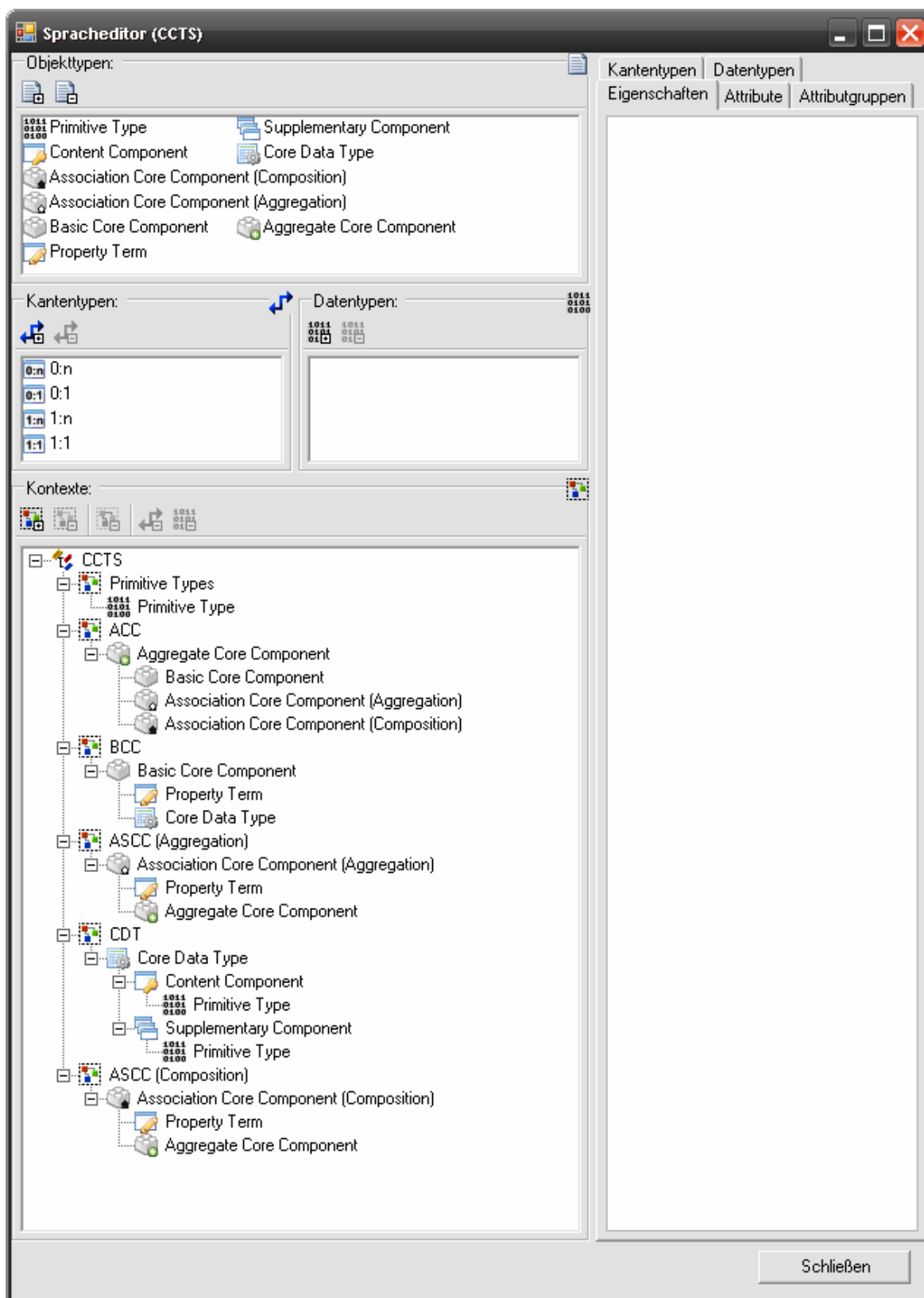


Abb. 27: Objekttypen und Kontexte der CCTS im Spracheditor

3.2.2 Modellierung der UBL

Das ERM der UBL ist nahezu identisch zu der rechten Seite des ERMs der CCTS (vgl. Abb. 28). Die Struktur der Business Information Entities unterscheidet sich nur in den zusätzlich modellierten Hierarchien. Da in UBL generische BIE statt Core Components für die Variantenerstellung existieren, drückt diese Hierarchie die Ableitungsbeziehung zwischen untergeordneten und übergeordneten BIE aus. Der ET *Data Type* (DT) besitzt ebenfalls eine Hierarchie, um die Beziehung zwischen generischen und adaptierten Datentypen auszudrücken, da in UBL sowohl qualifizierte als auch unqualifizierte Datentypen verwendet werden können. Technisch werden in UBL Supplementary Components eines DTs als Attribute, und die Content Component als Wert des DTs modelliert. Da die Namensgebung der Komponenten innerhalb des XML-Schemas unterschiedlich zu dem CCTS-Schema `Object Class.Property Term.Representation Term` ist, muss es bei der Modellierung explizit berücksichtigt werden, um die Übertragbarkeit zwischen H2-Modellen und XML-Schemata zu ermöglichen. Das Adaptionprinzip über Kontextkategorien und -ausprägungen ist in UBL identisch zu dem der CCTS, und besitzt daher auch die gleichen Entitytypen und Relationen.

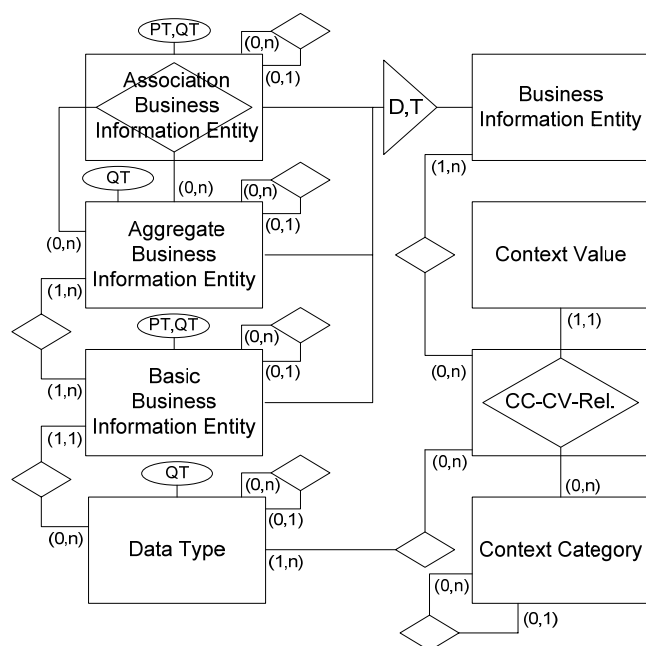


Abb. 28: ERM des UBL-Konzepts

Wie bei dem CCTS-Ansatz ist auch das ERM des UBL-Konzepts an die Gegebenheiten des H2-Toolsets zu adaptieren (vgl. Abb. 29). Dabei wird der für die Konfiguration ursprünglich relevante Teil (ET *Context Value* und *Context Category* sowie der uminterpretierte ET *CC-CV-Rel.*) durch den ET *Configuration Term* ersetzt. Wie bei der Modifikation

der CCTS ändern sich auch hier die Kardinalitäten entsprechend der Restriktion des H2-Toolsets, dass ein Element maximal mit einem Term belegt werden darf.

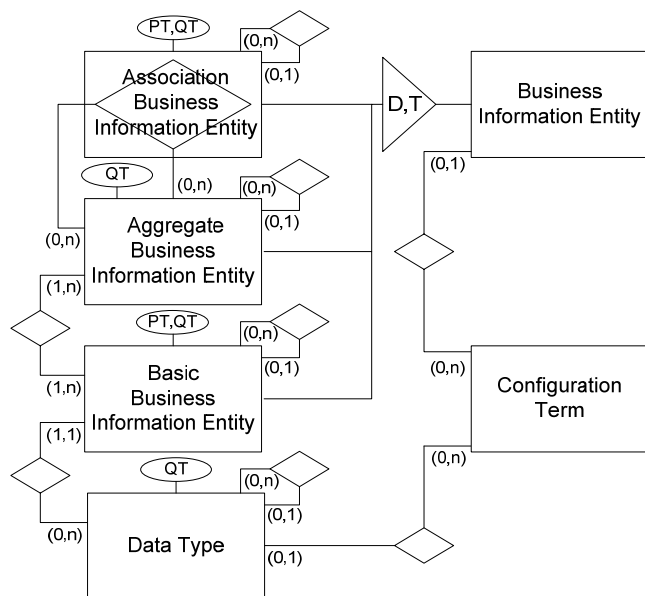


Abb. 29: ERM des adaptierten UBL-Konzepts für das H2-Toolset

Die Modellierung der UBL-Sprache erfolgt größtenteils identisch zu jener der CCTS. Analog zu den Objekttypen der Core Components in CCTS werden für UBL Business Information Entities und entsprechende Kontexte definiert (vgl. Abb. 30).

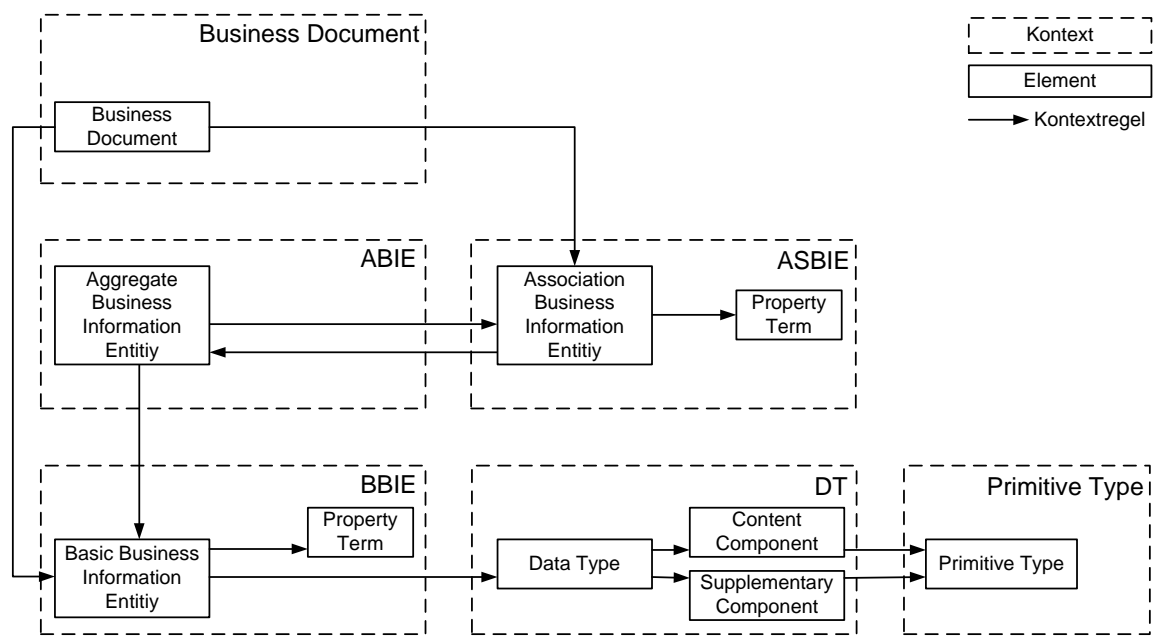


Abb. 30: Kontext- und Elementstruktur der UBL

In UBL wird ergänzend der Objekttyp *Business Document* und ein gleichnamiger Kontext eingeführt. Ein Business Document gleicht semantisch einem ABIE, enthält jedoch auf Modellebene nur die in der UBL definierten Geschäftsdokumente. Somit wird die Über-

sichtlichkeit der Modelle erhöht, da klar ersichtlich ist, welche Dokumente auf der obersten Hierarchieebene angeordnet sind. Strukturell nimmt der Kontext Business Document eine *aggregierende* Rolle ein, da die Komponenten der anderen Kontexte in diesem zu einem Wurzelknoten innerhalb des Modells zusammengefasst werden.

Die Modellierung von Sprachen und zugehörigen Modellen im H2-Toolset erfolgt implementierungsunabhängig. Die in der CCTS formulierte Anforderung nach Syntaxneutralität ist somit gewährleistet. Um die erstellten Modelle und Modellvarianten von UBL jedoch in der Praxis zu nutzen, sind diese zuvor in XML-Schemata zu übersetzen. Diese Übersetzung wird vom H2-Toolset aktuell noch nicht unterstützt. Jedoch könnte der bereits implementierte XML-Export durch ein sprachspezifisches Plug-In entsprechend erweitert und modifiziert werden. Um diesen XML-Export zu ermöglichen, sind die H2-Modelle durch Attribute um zusätzliche Informationen zu erweitern. Eine Analyse der XML-Schemata der UBL-Geschäftsdokumente hat ergeben, dass ein Großteil der für den Export benötigten Informationen bereits aus der Struktur der H2-Modelle extrahierbar ist. Neben dem Attribut *Restriction/Restriction Value*, das für die Restriktion der Datentypen erforderlich ist, werden im H2-Toolset folgende Attribute modelliert, um die für den XML-Export benötigten Informationen zu vervollständigen (vgl. Abb. 31):¹⁷⁴

- Das Attribut *Header* enthält syntaxspezifische Zusatzinformationen eines Geschäftsdokuments, wie z. B. verwendete Namespaces innerhalb des XML-Schemas.
- Das Attribut *Alternative Business Text* ermöglicht die Definition alternativer Bezeichnungen für Business Information Entities, wie z. B. „*Order Acknowledgement*“ statt *Order Response Simple*.
- Im Attribut *Definition* können die Definitionen von Business Information Entities gespeichert werden, wie z. B. „*The document used to indicate simple acceptance or rejection of an entire Order*“ für das Dokument *Order Response Simple*.
- Das Attribut *Version* enthält die Versionsnummer des jeweiligen Business Information Entities. Die Zuordnung von Versionsnummern ist wichtig, um Kompatibilität während des Austausches von Geschäftsdokumenten zu gewährleisten, da UBL sich in einem ständigen Entwicklungsprozess befindet.
- Das Attribut *Default-Wert* trägt dem Umstand Rechnung, dass qualifizierte Datentypen in UBL mit einem vordefinierten Wert versehen sind. Die qualifizierten Datentypen werden als Konsequenz nicht als eigener Objekttyp modelliert, da der Un-

¹⁷⁴ Die Zuweisung von Attributen zu Objekttypen im Spracheditor konnte mit der aktuellen Version des H2-Toolsets aufgrund technischer Probleme nicht realisiert werden.

terschied eben nur in der Spezifikation eines Default-Wertes liegt. Modellierungstechnisch wäre es jedoch ebenso möglich, qualifizierte Datentypen als eigenen Objekttypen zu modellieren.

- Das Attribut UBL-Name enthält die Alternativbezeichnung für Data Types, da diese innerhalb des XML-Schemas andere Namen erhalten als das CCTS-Schema vorsieht.

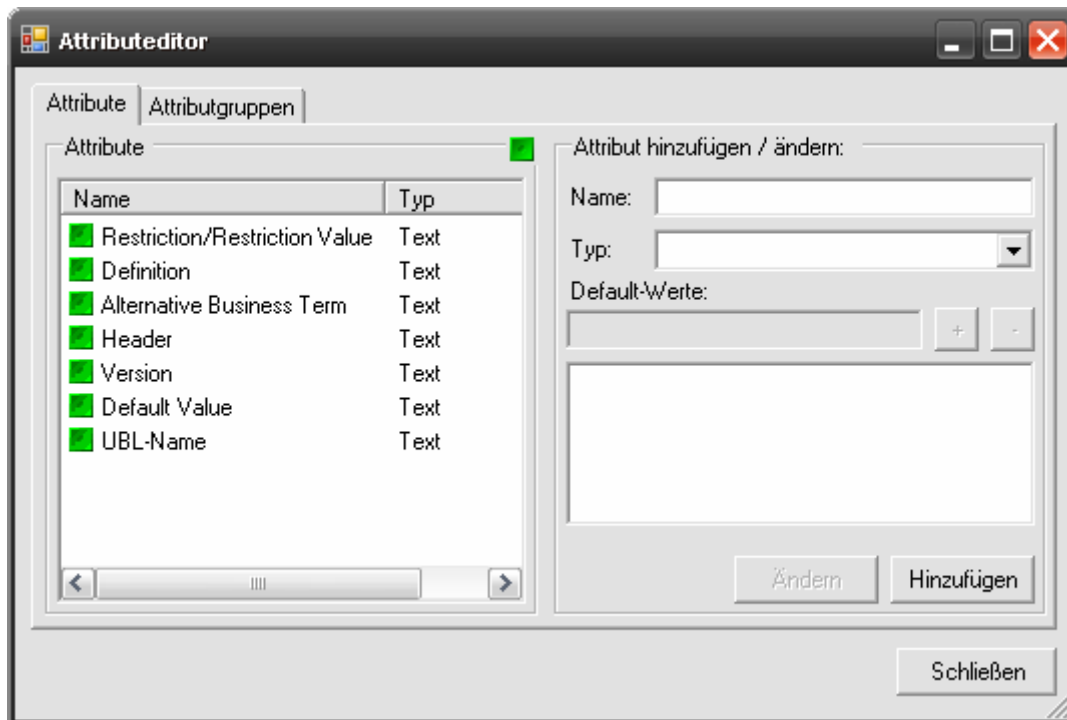


Abb. 31: Modellierte Attribute für Objekttypen der UBL

Ein Überblick über die Objekttypen, Kantentypen, Beziehungen und Kontexte der UBL-Sprache ist Abb. 32 zu entnehmen.

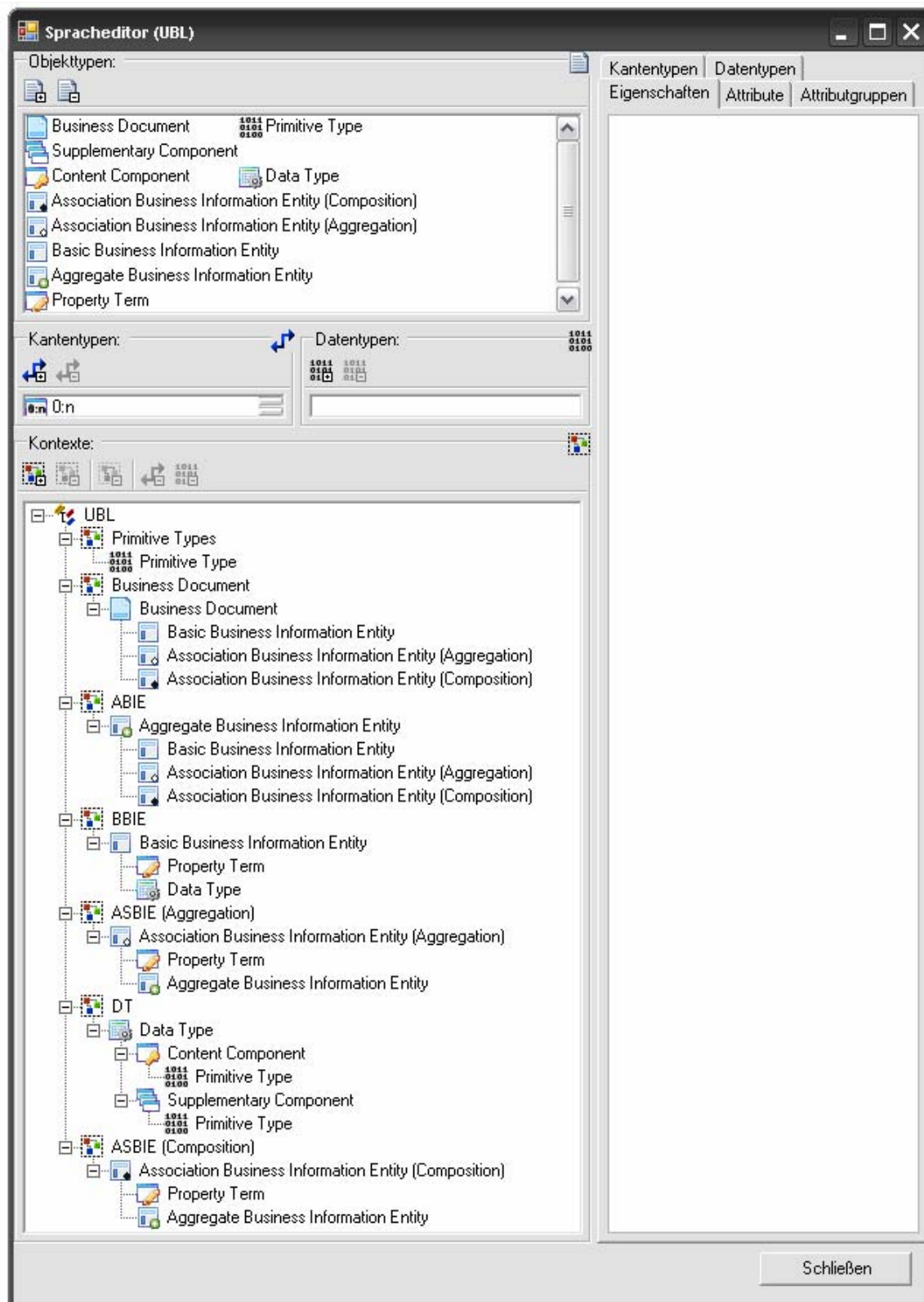


Abb. 32: Objekttypen und Kontexte der UBL im Spracheditor

3.3 Exemplarische Umsetzung zweier Geschäftsdokumente

3.3.1 Modellierung der ACC Address in CCTS

Zum aktuellen Zeitpunkt existiert für die CCTS keine offizielle Library für Geschäftsdokumente. Ein solches Repository ist jedoch für sämtliche Interoperabilitätsbestrebungen unerlässlich. Das H2-Toolset bietet daher die Möglichkeit, ein derartiges Repository als Modelle einer Geschäftsdokumentsprache zu erstellen und stellt somit eine Alternative zu offiziellen Veröffentlichungen dar.

Ein Geschäftsdokument ist letztlich nur eine Aggregate Core Component auf oberster Hierarchieebene. Zur Verdeutlichung des Modellierungskonzepts im H2-Toolset erscheinen die vordefinierten ACC der Core Component Library daher ebenso geeignet. Exemplarisch wurde ein Modell der ACC `Address` erstellt. Diese ACC repräsentiert den Standort, an dem eine bestimmte Organisation oder Person erreicht werden kann. Bei der Modellierung ist eine strukturierte Vorgehensweise zu präferieren. Ein Top-Down-Ansatz könnte folgendermaßen aussehen:

1. Erstellung Root-Elements `Address` im Kontext ACC.
2. Identifikation der zugehörigen BCC und ASCC innerhalb der Core Component Library.
3. Modellierung der BCC im gleichnamigen Kontext. Annotation der Property Terms unter die BCC. Erstellung einer Occurrence der BCC unter der zugehörigen ACC und Spezifikation der Kardinalitäten (vgl. Abb. 33).

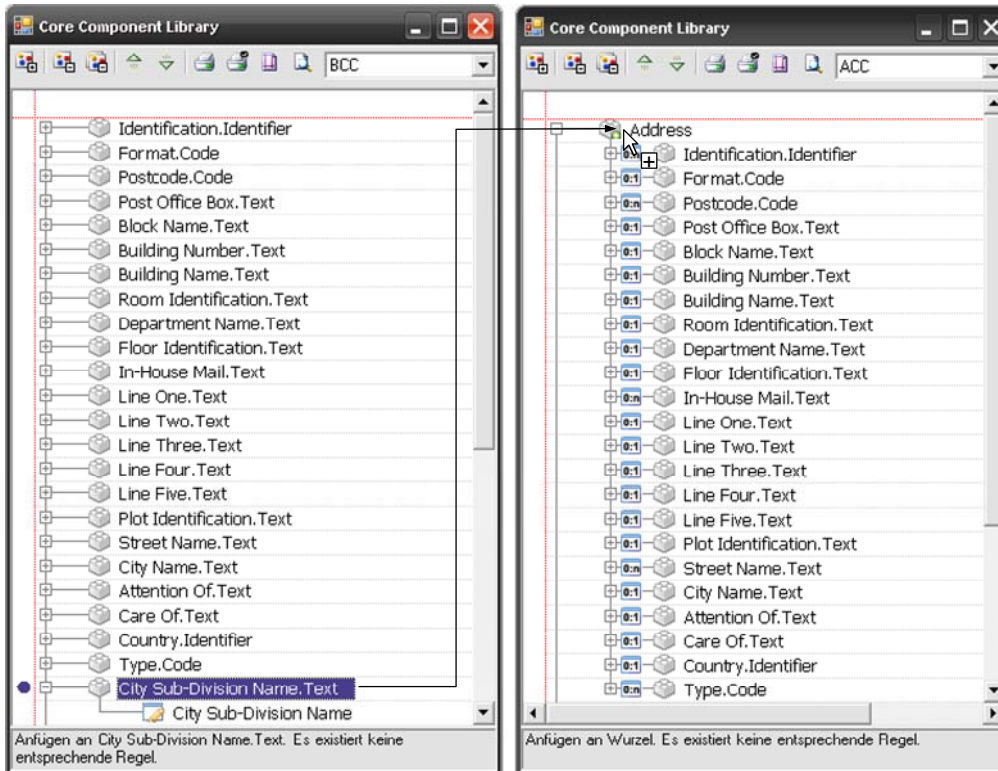


Abb. 33: Modellierung von BCC

4. Modellierung der ASCC im gleichnamigen Kontext. Annotation der Property Terms unter die ASCC. Erstellung einer Occurrence der ASCC unter der zugehörigen ACC und Spezifikation der Kardinalitäten.
5. Zu jeder ASCC ist die entsprechende ACC im Kontext ACC anzulegen und eine Occurrence dieser ACC als impliziter Representation Term unter die ASCC zu ziehen (vgl. Abb. 34).

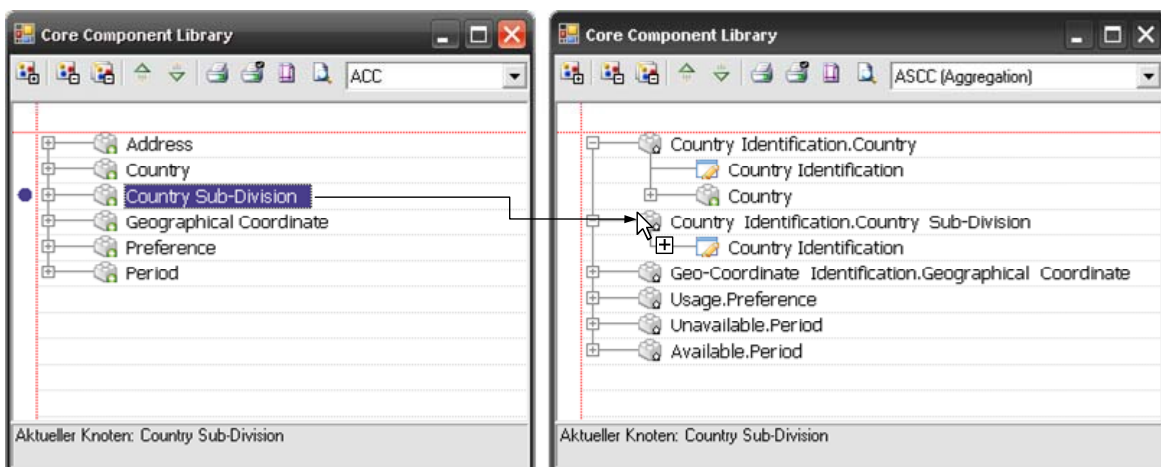


Abb. 34: Annotation einer ACC-Occurrence unter eine ASCC

6. Zu diesen neu angelegten ACC sind in einem rekursiven Prozess wieder die Schritte 2 bis 5 zu durchlaufen, bis alle BCC und ASCC modelliert sind.
7. Daraufhin sind die verwendeten Datentypen innerhalb der BCC zu analysieren. Dem Ergebnis entsprechend werden die Datentypen im Kontext CDT als Core Data Types angelegt und als Occurrences unter die BCC gehangen. Den Core Data Types werden entsprechend UN/CEFACT (2006c) Content- und Supplementary Components sowie zugehörige Kardinalitäten annotiert.
8. In einem letzten Schritt werden die Content- und Supplementary Components auf ihre primitiven Datentypen untersucht. Diese werden im Kontext Primitive Type erstellt und als Occurrences unter die jeweiligen Content- und Supplementary Components gehangen (vgl. Abb. 35).

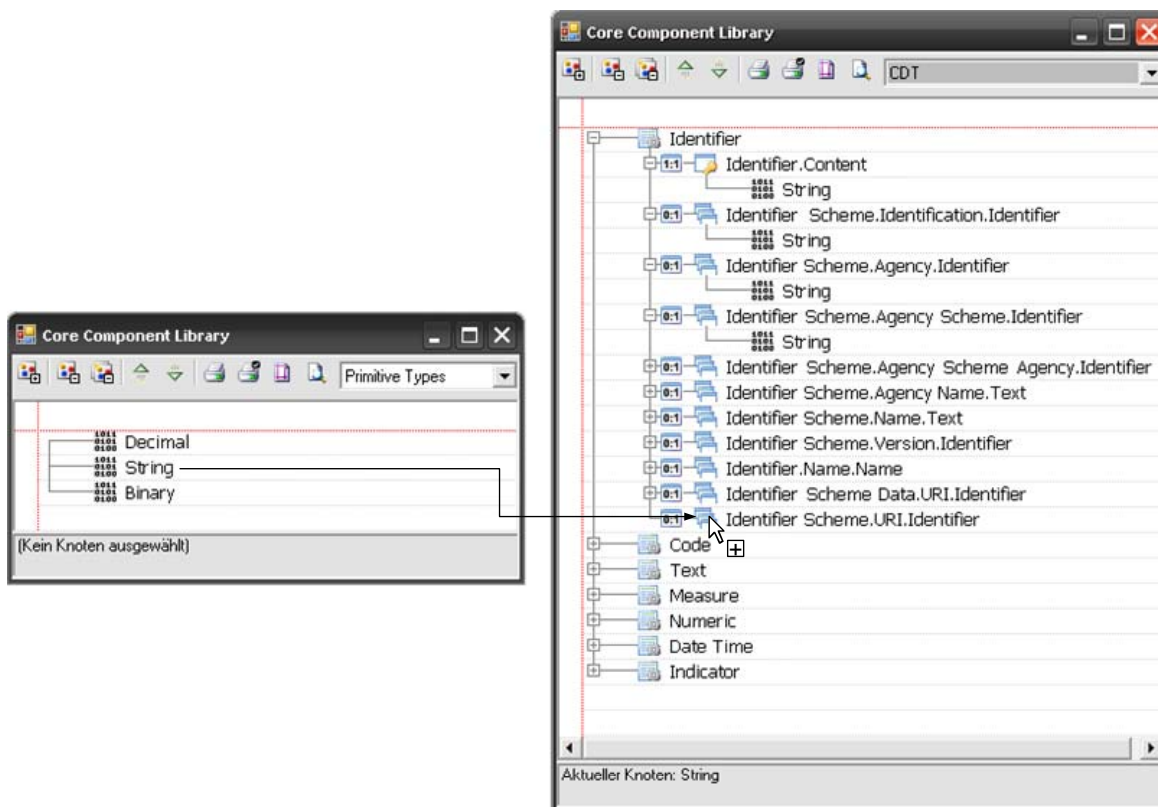


Abb. 35: Modellierung von Primitive Types und Core Data Types

Dieser relativ triviale Modellierungsprozess kann äußerst unübersichtlich werden, da die Komponenten teilweise tief verschachtelt sind. Durch die Unübersichtlichkeit steigt die Gefahr, Komponenten und ihre Strukturen im H2-Toolset fehlerhaft zu modellieren. Aus diesem Grund wäre die Implementierung einer Importschnittstelle für XML-Schemata bzw. eines erweiterten Excelimports im H2-Toolset wünschenswert, mit dem die in den Dateien enthaltenen Modelle automatisch eingelesen und erstellt werden können.

Ein Anhaltspunkt über die Komplexität selbst relativ trivialer ACC ist Tab. 5 zu entnehmen. Die Tabelle enthält die Anzahl der Objekttypen, die für die Erstellung der ACC Address nötig sind. Die Anzahl der Objekttypen bezieht sich dabei nur auf die Definitionen der Elemente. Die Häufigkeiten von Occurrences dieser Definitionen wurde nicht berücksichtigt, da diese an Aussagekraft verlieren, sobald bewusste Endlosschleifen innerhalb einer Komponente auftreten. Die hohe Anzahl zu definierender Elemente für diese Aggregate Core Component verdeutlicht den enormen Aufwand, den die Erstellung einer Modellvariante im traditionellen CCTS-Ansatz verursacht. Für das Design einer Geschäftsdokumentvariante wären sowohl technische als auch betriebswirtschaftliche Kapazitäten einzubinden, die über die Relevanz und Implementierung jedes einzelnen Elementes zu entscheiden hätten. Dies ist gerade für kleine und mittlere Unternehmen problematisch, da derartige Kapazitäten dort oft nicht oder nur unzureichend vorhanden sind.

ACC Address	Anzahl Elemente
Aggregate Core Components	5
Association Core Components	6
Basic Core Components	40
Property Terms	46
Core Data Types	7
Content Components	7
Supplementary Components	39
Primitive Types	3
Gesamt	153

Tab. 5: Übersicht über die in Address enthaltenen Elemente und Häufigkeiten

Ein Überblick über das komplette Modell der ACC Address ist in Abb. 36 dargestellt.

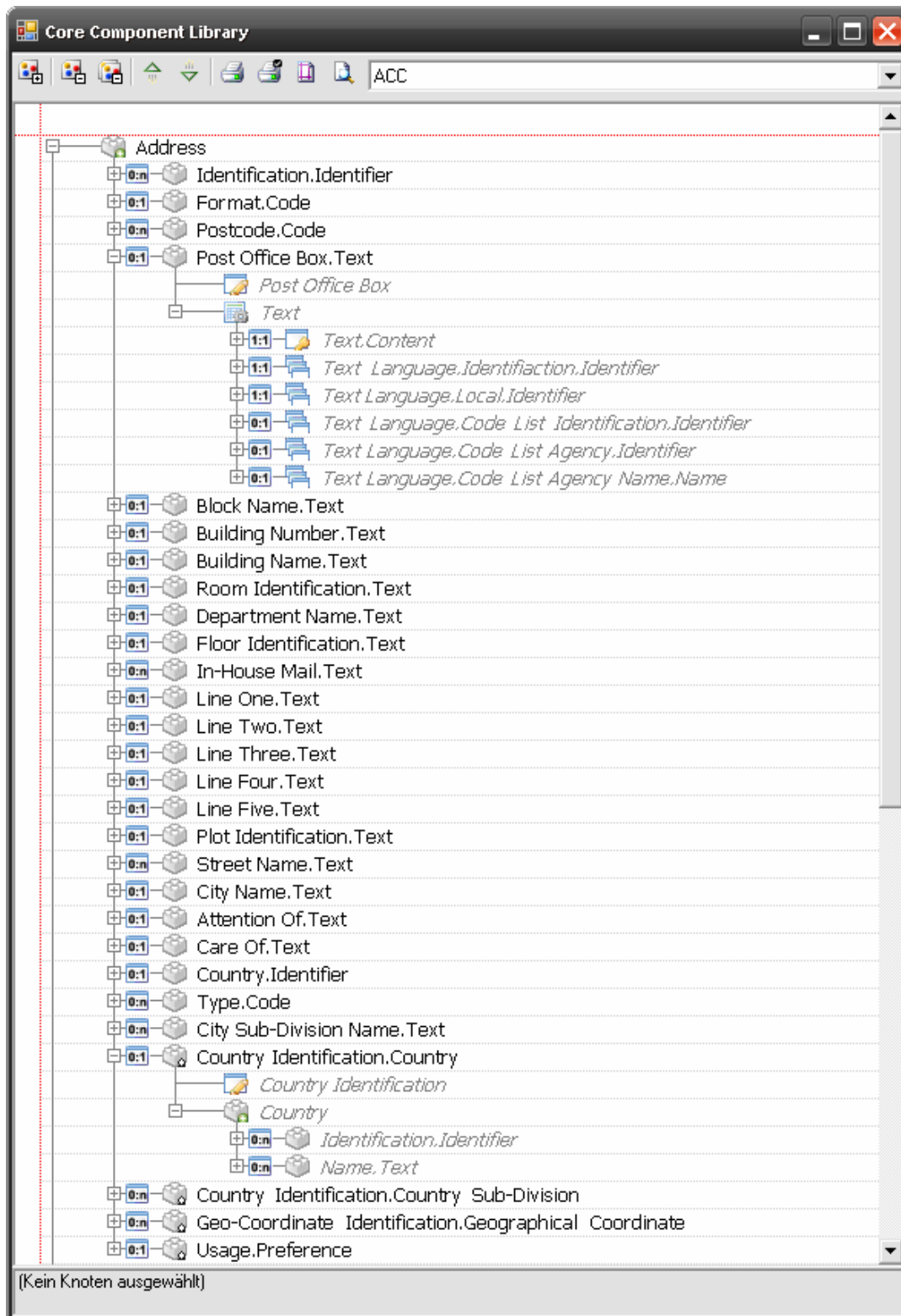


Abb. 36: ACC Address der Core Component Library im H2-Toolset

3.3.2 Modellierung des Geschäftsdokuments Order Response Simple in UBL

Die Modellierung eines Geschäftsdokuments in UBL unterscheidet sich nur unwesentlich von der in CCTS; einzig die Existenz von Business Information Entities statt Core Components und der zusätzliche Objekttyp Business Document sind bei UBL zu beachten. Das Wurzelement des Geschäftsdokuments wird dabei in dem Kontext Business Document angelegt und bildet die Ausgangsbasis für die sukzessive Top-Down-Modellierung nachfolgender Elemente. Da die Supplementary Component der DTs in UBL-Schemata als Attribute modelliert und dadurch keine Kardinalitäten vorgegeben sind, wurde die Annahme getroffen, diese auf (0:1) festzulegen, da ein Attribut entweder spezifiziert sein kann oder nicht. Der weitere Modellierungsprozess ist identisch zu dem bei der CCTS beschriebenen (vgl. Kap. 3.3.1).

Bei der Modellierung fiel die Wahl auf das Geschäftsdokument Order Response Simple. Es wird bei einer einfachen Bestätigung oder Verweigerung einer Bestellung versandt (vgl. Abb. 10 in Kap. 2.4.3) und stellt ein weniger umfangreiches Geschäftsdokument innerhalb der UBL-Bibliothek dar. Selbst triviale Geschäftsdokumente wie dieses enthalten jedoch eine sehr hohe Anzahl verschiedener Elemente (vgl. Tab. 6). Dabei bezieht sich die Anzahl der Objekttypen wiederum nur auf die Definitionen der Elemente. Neben der Menge trägt auch die tiefe Verschachtelung der Elemente zur Komplexität des Dokuments bei. Diese Kombination verdeutlicht die Problematik hinsichtlich benötigter Kapazitäten bei der Variantenerstellung, wie sie bereits im vorangegangenen Kapitel aufgezeigt wurden.

Order Response Simple	Anzahl Elemente
Aggregate Business Information Entities	24
Association Business Information Entities	40
Basic Business Information Entities	104
Property Terms	144
Core Data Types	9
Content Components	9
Supplementary Components	25
Primitive Types	3
Gesamt	358

Tab. 6: Übersicht über die in Order Response Simple enthaltenen Elemente und Häufigkeiten

Ein Überblick über das komplette Modell von Order Response Simple ist in Abb. 37 dargestellt.

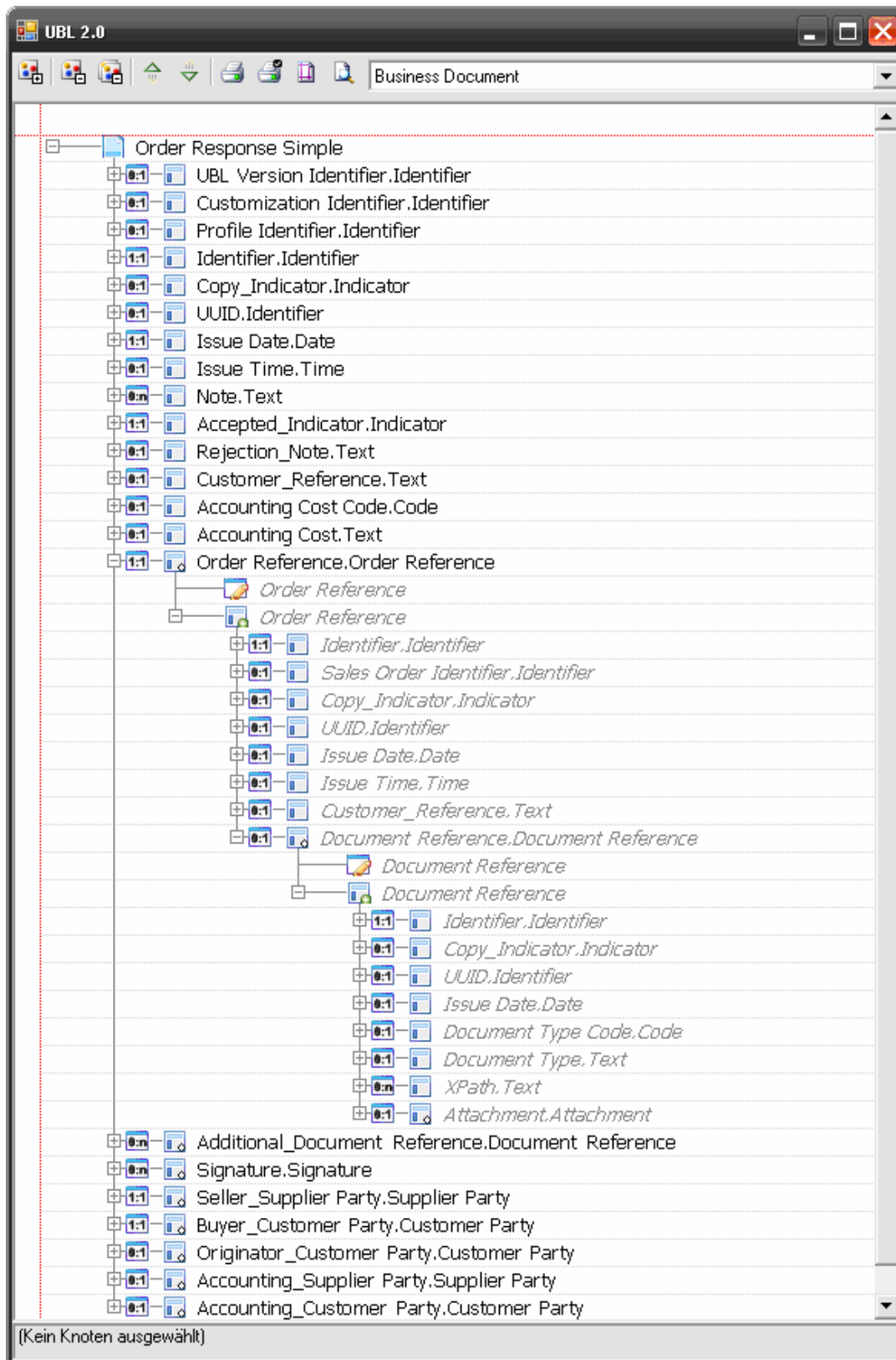


Abb. 37: Business Document Order Response Simple im H2-Toolset

3.4 Einordnung der Kontextkategorien in die Konfigurationsparameterstruktur der konfigurativen Referenzmodellierung

Die Begriffswelt der konfigurativen Referenzmodellierung weist für Konfigurationsparameter eine unternehmensfokussierte Betrachtungsweise auf. Unternehmensmerkmale und deren Ausprägungen stellen Eigenschaften dar, hinsichtlich derer *einzelne Unternehmen* klassifiziert werden können. Perspektiven wiederum repräsentieren spezifische Sichten auf das Referenzmodell, die durch die agierenden Nutzer *innerhalb* eines Unternehmens determiniert sind. Diese Sichtweise resultiert aus der ursprünglichen Anwendungsdomäne der konfigurativen Referenzmodellierung, die durch die Adaption von Referenzmodellen an die Bedürfnisse *eines* individuellen Unternehmens gekennzeichnet ist.

Im Gegensatz dazu liegt der Fokus elektronischer Geschäftsdokumente auf dem Austausch von Informationen zwischen Unternehmen. Bei der Konfiguration elektronischer Geschäftsdokumente sind diese hinsichtlich des Geschäftsprozesses sowie der beteiligten Unternehmen mit ihren unterschiedlichen Eigenschaften zu konfigurieren. Zudem sind bei der Konfiguration spezifische Perspektiven innerhalb eines Geschäftsprozesses zu berücksichtigen. So lassen sich bei dem Bestellprozess in UBL bspw. die Rollen *Buyer* und *Seller* unterscheiden. Diese passen nicht in die klassische Begriffswelt der konfigurativen Referenzmodellierung, da sie statt innerhalb eines Unternehmens innerhalb eines Geschäftsprozesses auftreten, in den unterschiedliche Unternehmen involviert sind. Das Referenzmodell von *Order Response Simple* und alle weiteren Geschäftsdokumente können somit ebenfalls aus unterschiedlichen Perspektiven betrachtet werden, die durch subjektive Sichten innerhalb eines Geschäftsprozesses determiniert sind.

Die Integration zwischen konfigurativer Referenzmodellierung und elektronischen Geschäftsdokumenten erfordert somit neben den technischen Aspekten, die durch das H2-Toolset bereitzustellen oder in dieses zu implementieren sind, auch theoretische Erweiterungen. Für den Kontext elektronischer Geschäftsdokumente wird daher eine abstraktere Definition von Unternehmensmerkmalen und Perspektiven vorgeschlagen. *Perspektiven* könnten im Kontext elektronischer Geschäftsdokumente als *subjektive Sichten definiert werden, die innerhalb eines Geschäftsprozesse eingenommen werden* (wie z.B. Ausprägungen von *Business Process Role* oder *Supporting Role*). *Unternehmensmerkmale* könnten die *Eigenschaften der in dem Geschäftsprozess involvierten Unternehmen* darstellen. Die Definition ist jedoch nicht optimal, da in einem Geschäftsprozess Eigenschaften auftreten können, die nicht durch die beteiligten Unternehmen determiniert sind. Daher wird die Definition auf *Eigenschaften des Geschäftsprozesses und der involvierten Unternehmen* erweitert. Diese Definition stimmt zwar mit der Bezeichnung Unternehmensmerkmal nicht mehr überschneidungsfrei überein, trägt aber zur Erfassung der Kontextkategorien

bei. Es wird deutlich, dass die Begriffswelten von konfigurativer Referenzmodellierung und CCTS bzw. UBL nicht verlustfrei ineinander überführbar sind. Die daraus resultierende, teils unpräzise Klassifizierung der Kontextkategorien der CCTS bzw. UBL in die Konfigurationsparameterstruktur der konfigurativen Referenzmodellierung stellt ein Defizit des aktuellen Ansatzes dar.

Letztlich wäre eine einfache Kombination der in UBL definierten Konfigurationskontextkategorien mit den Mechanismen der konfigurativen Referenzmodellierung ohne deren Einordnung in das Konzept von Unternehmensmerkmalen und Perspektiven möglich. Diese Alternative ist im H2-Toolset momentan nicht vorgesehen, könnte jedoch mit relativ geringem Aufwand implementiert werden, da die Funktionalität der Konfigurationsmechanismen nicht an diese beiden Kategorien gebunden ist. Dabei wäre das starre System von Unternehmensmerkmalen und Perspektiven gegen ein dynamisches zu ersetzen, das die Definition eigener Konfigurationskategorien ermöglicht. Diese Lösung wäre einer theoretisch sauberen Abbildung von Konfigurationskontextkategorien zuträglich.

Um in der aktuellen Version des H2-Toolsets dennoch eine Konfiguration der Referenzmodelle vorzunehmen, sind die Kontextkategorien der CCTS bzw. UBL in die erweiterte Definition von Unternehmensmerkmalen und Perspektiven einzuordnen (vgl. Tab. 7).

Klassifikationsschema	Unternehmensmerkmal	Perspektive
Business Process	√	
Product Classification	√	
Industry Classification	√	
Geopolitical Context	√	
Official Constraints	√	
System Capabilities	√	
Business Process Role		√
Supporting Role		√

Tab. 7: Klassifikationsschema der Kontextkategorien

Die Unterscheidung dient dabei nur der logischen Strukturierung. Für die Funktionalität der Konfigurationsmechanismen ist es irrelevant, ob eine Kategorie als Unternehmensmerkmal oder Perspektive klassifiziert wird. Dies liegt in der Tatsache begründet, dass im Unterschied zu dem Ansatz nach BECKER ET AL. im H2-Toolset keiner der Konfigurationsmechanismen nur an Unternehmensmerkmale oder Perspektiven gebunden ist, sondern die Konfiguration mit beiden Kategorien äquivalent durchführbar ist.

Die Kategorie *Business Process* spezifiziert den Geschäftsprozess des Unternehmens und die darin subsumierten Aktivitäten. Sie wurde als „Unternehmensmerkmal“ klassifiziert,

da die Bezeichnung des Geschäftsprozesses eine Eigenschaft desselben darstellt. Die Kategorie *Product Classification* umfasst Produkte bzw. Dienstleistungen, die während eines Geschäftsprozesses ausgetauscht bzw. modifiziert werden. Diese Produkte und Dienstleistungen stellen Eigenschaften des Geschäftsprozesses und der beteiligten Unternehmen dar, und werden daher als Unternehmensmerkmal klassifiziert. Selbiges gilt für die Kategorie *Industry Classification*, die die Industrie bzw. den Industriezweig spezifiziert, in dem der Geschäftsprozess stattfindet. *Geopolitical Context* subsumiert territoriale Eigenschaften auf unterschiedlicher Hierarchieebene sowie geographisch bedingte, kulturelle Faktoren, die den Geschäftsprozess beeinflussen. *Official Constraints* beinhaltet gesetzliche und regulatorische Anforderungen an den Geschäftsprozess während *System Capabilities* Systeme umfasst, die in diesen involviert sind. Die drei Kategorien sind somit ebenfalls der erweiterten Definition eines Unternehmensmerkmals zuzurechnen. Die Kategorien *Business Process Role* und *Supporting Role* umfassen rollenspezifische Faktoren und stellen eher Sichten innerhalb eines Geschäftsprozesses dar. Sie werden daher als Perspektiven modelliert. Ein Ausschnitt des Konfigurationsmodells im H2-Toolset ist in Abb. 38 visualisiert.

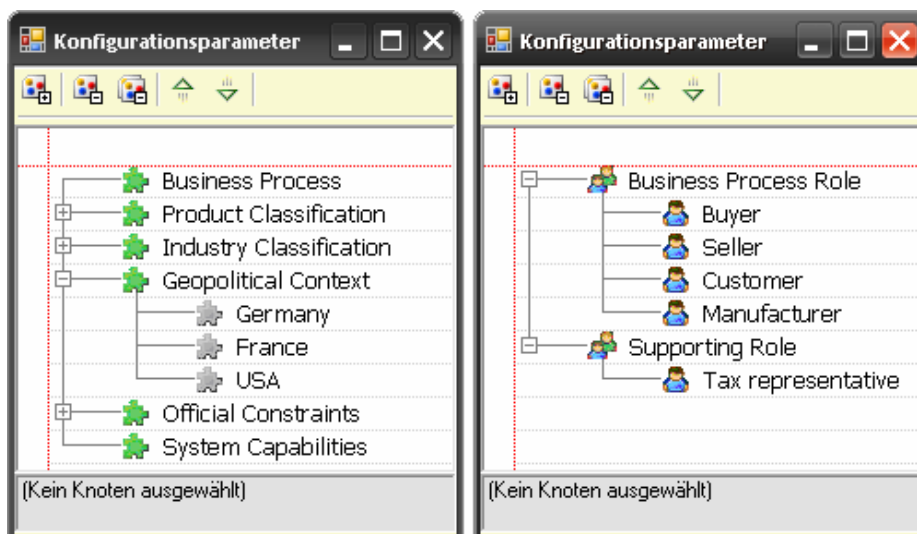


Abb. 38: Modell der Konfigurationsparameter von CCTS bzw. UBL im H2-Toolset

Einige Kontextkategorien der CCTS bzw. UBL erfordern mehrere Hierarchieebenen, die entsprechende Ausprägungen beinhalten können (bspw. die Hierarchie innerhalb von *Geopolitical Context*). Die Abbildung von Hierarchieebenen in der Konfigurationsstruktur des H2-Toolsets ist in der aktuellen Version nur zweistufig möglich: Auf der ersten Ebene wird das Unternehmensmerkmal bzw. die Perspektive definiert, auf der zweiten Ebene die entsprechenden Ausprägungen. Aufgrund dieser Restriktion ist bei der Modellierung abzuwägen, ob alle Hierarchieebenen einer Kontextkategorie direkt auf erster Ebene erstellt werden, oder ob eine aggregiertere Darstellung erfolgt. Um die Übersichtlichkeit zu erhalten, wurde in dieser Arbeit letztere Variante gewählt. Ausprägungen von Kontextkatego-

rien wie *Geopolitical Context* wurden dabei auf ein angemessenes Aggregationsniveau wie die Länderebene beschränkt.

Ein weiteres Defizit im H2-Toolset ist in der fehlenden ontologischen Unterstützung der Kategorien von Unternehmensmerkmalen und Perspektiven zu sehen. Bei der CCTS wird für manche Kontextkategorien gefordert, Hierarchiestufen komplett zu spezifizieren. Ein Beispiel hierfür ist die Kategorie *Geopolitical Context*, bei der, falls möglich, die entsprechenden Ausprägungen vom Kontinent bis zur Region zu spezifizieren sind.¹⁷⁵ Ein weiteres Beispiel könnte die Kategorie *Official Constraints* darstellen, die in einigen Fällen eine Ausprägung eines *geopolitischen Kontexts* erfordern kann (wie z. B. die Mitgliedschaft eines Freihandelsabkommens an ein spezifisches Land gebunden ist). Eine Erweiterung ontologischer Beziehungen im H2-Toolset von Konfigurationsparameterausprägungen auf komplette Kategorien erscheint daher wünschenswert.

3.5 Modellkonfiguration

3.5.1 Evaluation der Möglichkeiten anhand einer Fallstudie

Die Möglichkeiten der Konfiguration elektronischer Geschäftsdokumente mit dem H2-Toolset werden im Folgenden anhand einer Fallstudie evaluiert.

Ausgangssituation

Die Panther GmbH ist ein in Deutschland ansässiger Sportschuhproduzent. Diese vertreibt ihr Sortiment direkt an Großhändler über ein webgestütztes Informationssystem. Nach einmaliger Authentifizierung können dort Bestellungen aufgegeben werden. Um Gebühren zu sparen, erfolgt die Abrechnung direkt über die Panther GmbH und die teilnehmenden Kunden. Die Bestellungen wurden bislang intern weiterverarbeitet, und eine Bestätigung oder Ablehnung per E-Mail an die entsprechenden Händler zurückgeschickt. Um die Potenziale moderner EDI-Standards auszunutzen, entscheidet sich die Geschäftsführung für die Einführung von UBL als Austauschformat für den Bestellprozess. Die Daten sollen nach der Modernisierung nicht mehr manuell per E-Mail versendet werden, sondern über ein automatisiertes System, bei dem der Auftrag nur noch überprüft und gegebenenfalls bestätigt oder abgelehnt wird. Seit der WM 2006 hat die Panther GmbH auch in den USA einen großen Bekanntheitsgrad erreicht. Da das Marktpotenzial von der Geschäftsführung als bedeutend eingestuft wird, gibt es Bestrebungen, eine dortige Niederlassung zu grün-

¹⁷⁵ Vgl. UN/CEFACT (2006b), S. 70.

den. Der Handel soll dort in direktem Kontakt mit den entsprechenden Einzelhändlern erfolgen, um die Modalitäten möglichst einfach zu gestalten. Aus Sicherheitsgründen soll die Abrechnung in den USA über einen Drittanbieter abgewickelt werden, der mit den rechtlichen Bestimmungen besser vertraut ist. Restposten der letzten Saison werden über einen in Frankreich ansässigen Partnerhändler weitervertrieben, der seine Bestellungen über das Internet platziert. Die Mitarbeiter der Panther GmbH sind mit den Gegebenheiten von UBL nicht vertraut. Aus diesem Grund wird eine Unternehmensberatung hinzugezogen, die die entsprechende Konfiguration der Geschäftsdokumente vornehmen soll. Da UBL ein CCTS implementierender Standard ist, soll auch die Konfiguration letzterer kurz erläutert werden. Die Konfiguration erfolgt mit dem H2-Toolset und ist auf das Geschäftsdokument `Order Response Simple` beschränkt. Das Geschäftsmodell der Panther GmbH ist in Abb. 39 zusammengefasst.

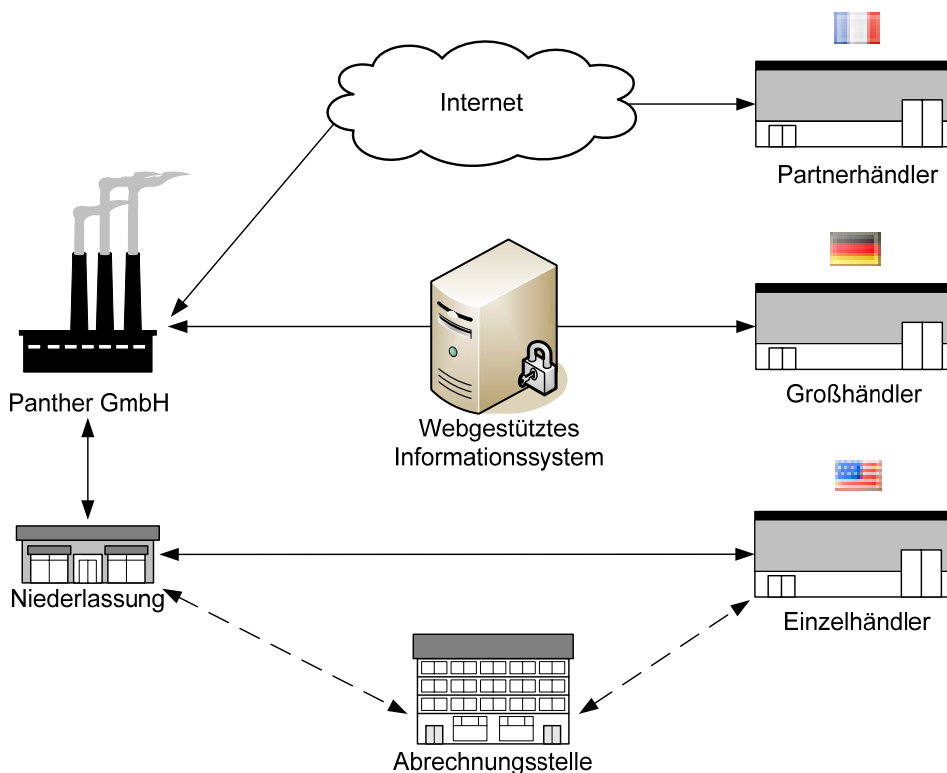


Abb. 39: Geschäftsmodell der Panther GmbH

Die Konfiguration des Geschäftsdokuments lässt sich in zwei Phasen unterteilen. In der ersten Phase sind die relevanten Konfigurationsparameter zu identifizieren und modellieren. Dabei sind mögliche Interdependenzen zwischen den Konfigurationsparameterausprägungen durch ontologische Beziehungen zu spezifizieren. In der zweiten Phase erfolgt die Top-Down-Konfiguration des Geschäftsdokuments. Dabei sind die Elemente innerhalb des Dokuments darauf zu untersuchen, ob sie in den identifizierten Konfigurationskontexten relevant sind oder nicht. Diese Beziehung ist durch Konfigurationsterme auszudrücken, die

den Elementen annotiert werden. Ergeben sich während der Konfiguration neue Erkenntnisse bzgl. der Beziehungen zwischen Konfigurationsparametern, so ist ein Rücksprung in die erste Phase möglich, um Ergänzungen vorzunehmen. Folgende Konfigurationsparameterausprägungen wurden identifiziert:

Kategorie	Ausprägungen
Business Process	Ordering
Product Classification	Shoes
Industry Classification	Apparel
Geopolitical Context	Germany, France, USA
Official Constraints	-
Business Process Role	Seller
Supporting Role	-
System Capabilities	Web-based IS, Internetshop

Tab. 8: Kontextkategorien der Geschäftssituation

Diese Kontextkategorien werden anschließend in der Konfigurationssprache des H2-Toolsets modelliert (vgl. Abb. 40):

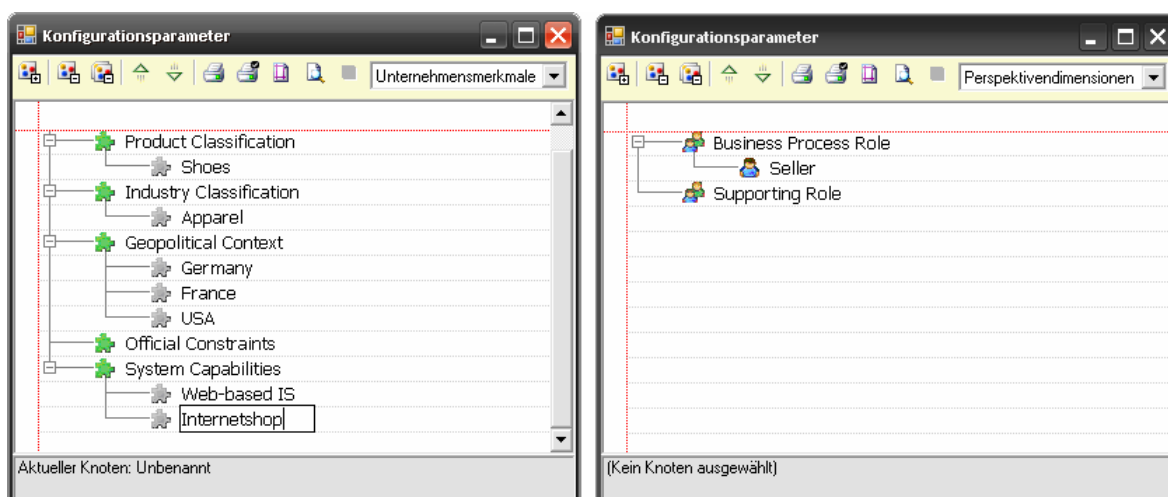


Abb. 40: Konfigurationsparameter der Geschäftssituation im H2-Toolset

Eine ontologische Beziehung wurde vorerst nicht festgestellt. Um die modellierten Konfigurationsparameter in dem Referenzmodell als relevant zu deklarieren, sind sie diesem zuzuordnen. Im Anschluss sind Terme für die Konfiguration zu definieren. Die Konfigurationsparameter werden dabei zu drei Gruppen zusammengefasst. Jede dieser Gruppen spezifiziert eines der drei möglichen Handelsszenarien der Puma GmbH. Die erste Gruppe subsumiert den Handel in Deutschland über das webbasierte Informationssystem. Die zweite Gruppe umfasst den Handel mit dem französischen Zwischenhändler, der über einen Internetshop abgewickelt wird. Die dritte Gruppe repräsentiert den direkten Handel mit amerikanischen Einzelhändlern, der durch die dortige Niederlassung abgewickelt werden soll. Zu diesen Gruppen werden entsprechende Konfigurationsterme formuliert (vgl. Abb.

41). Der Term *Basic* umfasst dabei die grundlegenden Eigenschaften, die für alle Handels-szenarien gleichermaßen gültig sind und die in diesen wiederverwendet werden. Innerhalb der Terme *Not (Information System Germany)*, *Not (Internetshop France)* und *Not (Direct sale USA)* wird der Term *Basic* mit einem logischen UND mit den Negationen der anderen Ausprägungen verknüpft. Die Negation ist erforderlich, da im Zuge der Konfiguration solche Elemente mit Termen belegt werden, die in den entsprechenden Kontexten *nicht* relevant sind und somit ausgeblendet werden. Bspw. sind Elemente, die mit dem Term *Not (Information System Germany)* belegt werden, irrelevant, wenn entweder die grundlegenden Eigenschaften des Handelsprozesses nicht erfüllt sind, oder die Konjunktion der speziellen Ausprägungen *Geopolitical Context is Germany* und *System Capabilities is Web-based IS* zutreffen.

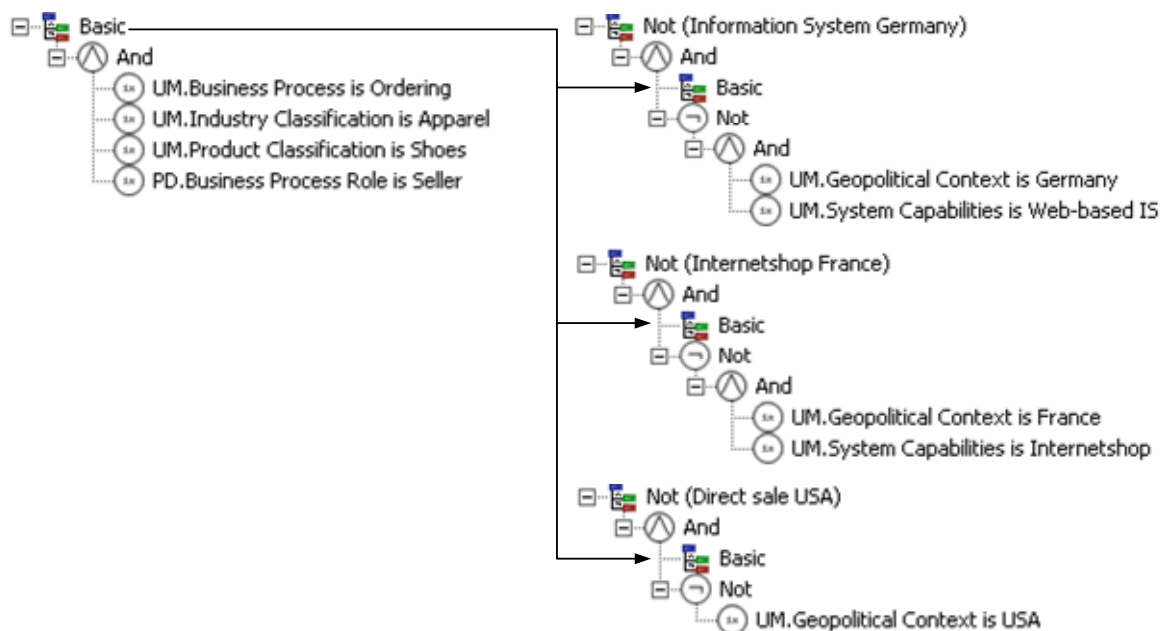


Abb. 41: Konfigurationsterme für einzelne Geschäftsszenarien

Da nur solchen Elementen Terme annotiert werden, die in einem spezifischen Kontext *ausgeblendet* werden sollen, ergeben sich Konsistenzprobleme bei der Auswahl von Konfigurationsparameterausprägungen: Werden für die Generierung einer spezifischen Variante die Konfigurationsparameterausprägungen nicht wie durch die Gruppenunterteilung vorgegeben gewählt, so kann dies zur Ausblendung von Elementen führen, die für eine Teilmenge der selektierten Konfigurationsparameter eigentlich relevant wären. Wäre bspw. ein Element X mit dem Term *Not (Information System Germany)* belegt und überlappende Konfigurationsparameter für eine Variante wie *Basic*, *Germany*, *France*, *Web-based IS* und *Internetshop* selektiert, so führte dies bei entsprechender Konfiguration zu einer Ausblendung des Elements X, obwohl dieses für eine Teilmenge der Konfigurationsparameter (*France* und *Internetshop*) relevant wäre. Für eine konsistente Modellierung ist es daher

essentiell, die zu erstellenden Varianten disjunkt auf die Anforderungen der Geschäftsszenarien zuzuschneiden. Dies erfordert einen Rücksprung in Phase eins, um entsprechende Bedingungen und Ausschlüsse der Konfigurationsparameterausprägungen zu definieren. Diese ontologische Spezifikation stellt sicher, dass bei der Variantenerstellung nur die vorgesehenen Kombinationen gewählt werden können. Dabei wurden die zusammengehörigen Kategorien durch zwei asymmetrische Bedingungen zu einer symmetrischen zusammengefasst. Im Anschluss wurde je eines dieser Merkmale innerhalb eines Geschäftsszenarios als abgrenzendes Kriterium gegenüber den beiden anderen Geschäftsszenarios als Ausschlussbeziehung formuliert (vgl. Abb. 42).¹⁷⁶

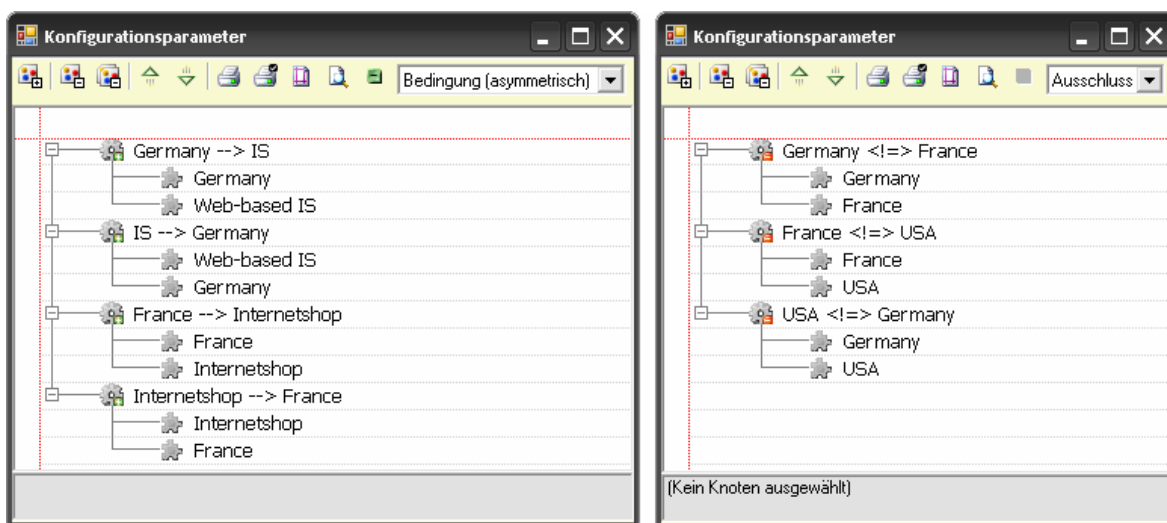


Abb. 42: Ontologische Beziehungen der Geschäftsszenarien

Die möglichen Kombinationen sind somit vollständig spezifiziert und ermöglichen im Folgenden die Generierung konsistenter Varianten. Nach der Definition von Termen und Ontologiebeziehungen kann das Geschäftsdokument *Order Response Simple* konfiguriert werden. Die folgenden Ausführungen beschränken sich auf die Elemente der obersten Ebene des Geschäftsdokuments. Zur Veranschaulichung der Konfigurationseinschränkungen wird zusätzlich das in allen ASBIE vom Typ *Party* vorkommende ABIE *Address* konfiguriert, da dort in einigen Fällen nur die gemeinsame Obermenge konfiguriert werden kann.

Grundlegende Konfiguration

Innerhalb des Geschäftsdokuments *Order Response Simple* konnten folgende Elemente identifiziert werden, die für *keines* der drei Geschäftsszenarien relevant sind (vgl.

¹⁷⁶ Eleganter wäre eine logische Verknüpfung der ontologischen Spezifikation wie im vorangegangenen Kapitel beschrieben.

Tab. 9). Um diese Elemente als irrelevant zu deklarieren, ist ein neuer Konfigurationsterm zu erstellen, der die einzelnen Konfigurationsterme der jeweiligen Geschäftsszenarien UND-verknüpft.

Name	Typ	Tiefe	Annahme
Additional_Document Reference. Document Reference	ASBIE	1	Zusätzliche Dokumente werden nicht benötigt
Originator_Customer Party. Customer Party	ASBIE	1	In den Szenarien entspricht der Auftraggeber dem Käufer
Copy_Indicator.Indicator	BBIE	1	Ausgetauschtes Dokument ist immer das Original
Customer_Reference.Text	BBIE	1	Zusätzlicher Referenztext wird nicht benötigt

Tab. 9: Irrelevante Elemente für alle Geschäftsszenarien

Im Anschluss ist dieser Term den Elementen im Modell zuzuweisen (vgl. Abb. 43).

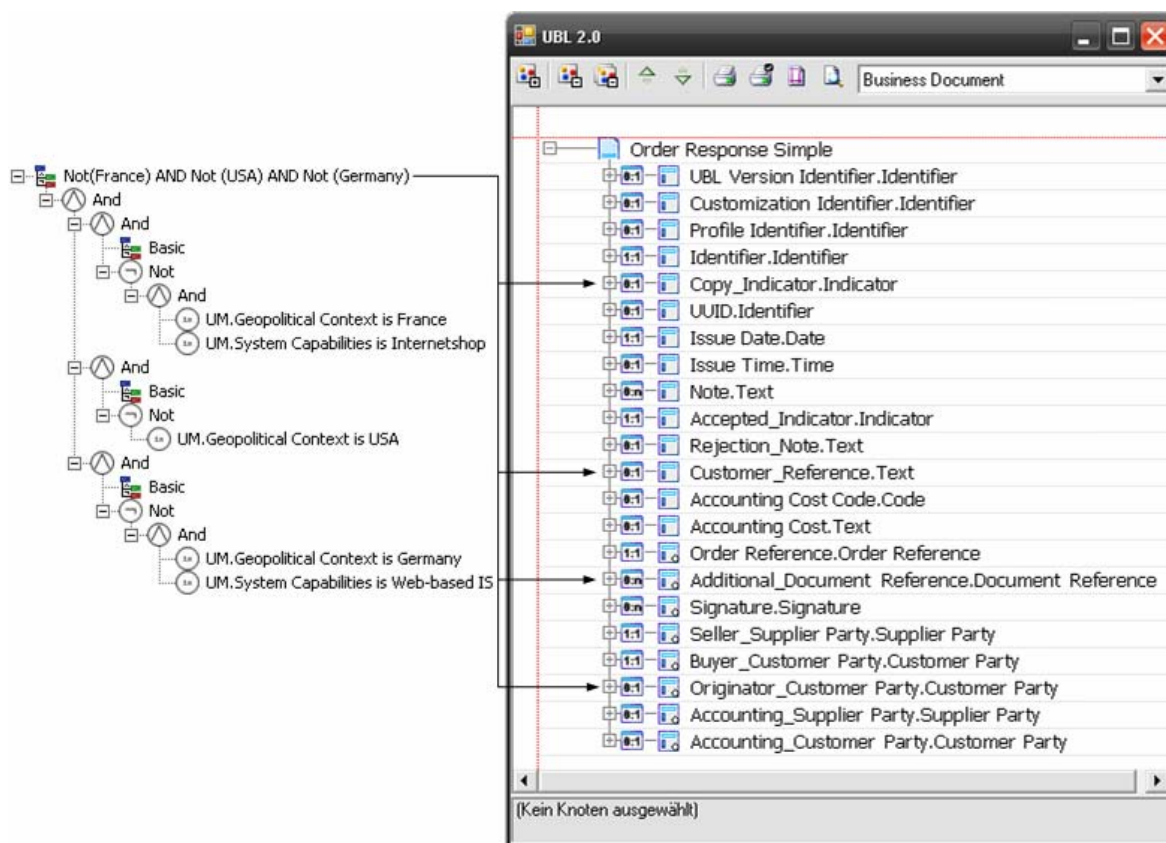


Abb. 43: Zuweisung des Terms zu irrelevanten Elementen aller Geschäftsszenarien

Handel mit deutschen Großhändlern über ein webbasiertes Informationssystem

Der Handel der Panther GmbH mit den deutschen Großhändlern erfolgt über ein webbasiertes Informationssystem. Die betroffenen Händler müssen sich einmalig authentifizie-

ren, und können im Anschluss Bestellungen über das Informationssystem aufgeben. Da die Kommunikation bzgl. des Bestellprozesses dadurch abgesichert ist, kann auf die Angabe einer Signatur verzichtet werden. Zudem erfolgt die Abrechnung der Bestellungen direkt über die Panther GmbH als Anbieter und die beteiligten Handelsunternehmen als Kunden. Zusätzliche Parteien, die für die Abrechnung zuständig sind, werden daher nicht benötigt. Da sowohl Anbieter als auch Kunden innerhalb von Deutschland ansässig sind, kann die Adresse den lokalen Anforderungen entsprechend konfiguriert werden. Folgende Elemente wurden für das skizzierte Geschäftsszenario als irrelevant identifiziert (vgl. Tab. 10).

Name	Typ	Tiefe	Annahme
Signature.Signature	ASBIE	1	Bestellungen und Nutzer sind bereits authentifiziert
Accounting_Supplier Party. Supplier Party	ASBIE	1	Abrechnung erfolgt direkt
Accounting_Customer Party. Customer Party	ASBIE	1	Abrechnung erfolgt direkt
Floor.Text	BBIE	7	für Deutschland irrelevant
Room.Text	BBIE	7	für Deutschland irrelevant
Additional_Street Name.Name	BBIE	7	für Deutschland irrelevant
Block Name.Name	BBIE	7	für Deutschland irrelevant
Building Name.Name	BBIE	7	für Deutschland irrelevant
Inhouse_Mail.Text	BBIE	7	für Deutschland irrelevant
Department.Text	BBIE	7	für Deutschland irrelevant
Mark Attention.Text	BBIE	7	für Deutschland irrelevant
Mark Care.Text	BBIE	7	für Deutschland irrelevant
Plot Identification.Text	BBIE	7	für Deutschland irrelevant
City Subdivision Name.Name	BBIE	7	für Deutschland irrelevant
Country Subentity.Text	BBIE	7	für Deutschland irrelevant
Country Subentity Code.Code	BBIE	7	für Deutschland irrelevant
Region.Text	BBIE	7	für Deutschland irrelevant
District.Text	BBIE	7	für Deutschland irrelevant
Timezone Offset.Text	BBIE	7	für Deutschland irrelevant
Address Line.Address Line	ASBIE	7	für Deutschland irrelevant
Country.Country	ASBIE	7	für Deutschland irrelevant
Location Coordinate. Location Coordinate	ASBIE	7	für Deutschland irrelevant

Tab. 10: Irrelevante Elemente für Handel mit deutschen Großhändlern über ein web-basiertes Informationssystem

Diesen Elementen ist der Term Not (Information System Germany) zuzuweisen, um sie im entsprechenden Kontext auszublenden (vgl. Abb. 44).

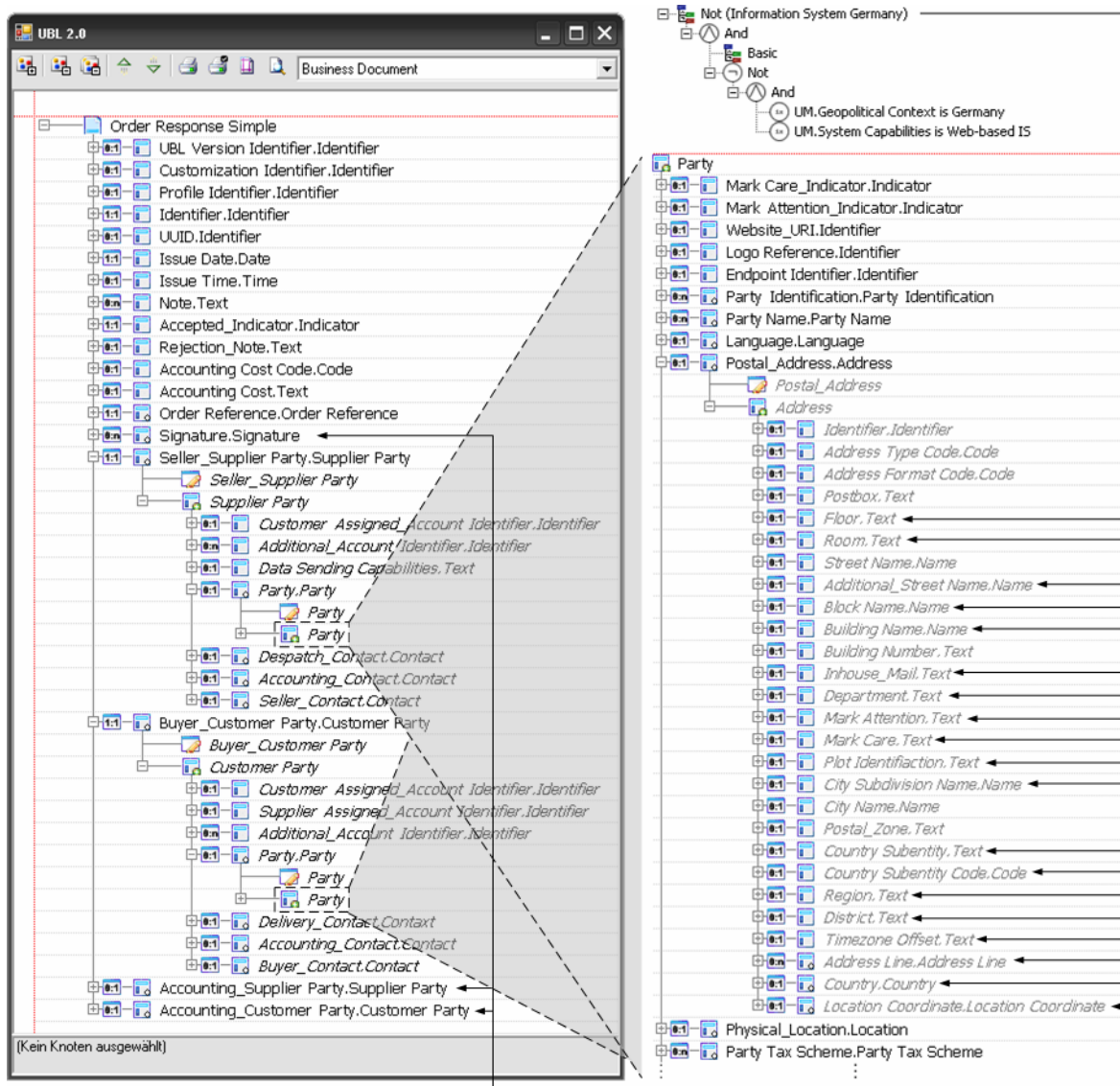


Abb. 44: Zuweisung von Termen für Handel mit deutschen Großhändlern über ein webbasiertes Informationssystem

Handel mit französischen Zwischenhändlern über einen Internetshop

Der Verkauf der Restposten an französische Zwischenhändler erfolgt über einen Internetshop. Die Kommunikation ist dabei nicht durch das beteiligte System sichergestellt, so dass die Verwendung einer Signatur weiterhin erforderlich ist. Die Abrechnung der Ware erfolgt wiederum direkt über die Panther GmbH und die beteiligten Zwischenhändler, um die Kosten für eine zusätzliche Abrechnungsstelle einzusparen. Diese Parteien sind daher für das konfigurierte Geschäftsdokument irrelevant. Folgende Elemente wurden für das skizzierte Geschäftsszenario als irrelevant identifiziert (vgl. Tab. 11).

Name	Typ	Tiefe	Annahme
Accounting_Supplier Party. Supplier Party	ASBIE	1	Abrechnung erfolgt direkt
Accounting_Customer Party. Customer Party	ASBIE	1	Abrechnung erfolgt direkt
Floor.Text	BBIE	7	für Frankreich irrelevant
Room.Text	BBIE	7	für Frankreich irrelevant
Additional Street Name.Name	BBIE	7	für Frankreich irrelevant
Block Name.Name	BBIE	7	für Frankreich irrelevant
Building Name.Name	BBIE	7	für Frankreich irrelevant
Inhouse Mail.Text	BBIE	7	für Frankreich irrelevant
Department.Text	BBIE	7	für Frankreich irrelevant
Mark Attention.Text	BBIE	7	für Frankreich irrelevant
Mark Care.Text	BBIE	7	für Frankreich irrelevant
Plot Identification.Text	BBIE	7	für Frankreich irrelevant
City Subdivision Name.Name	BBIE	7	für Frankreich irrelevant
Country Subentity.Text	BBIE	7	für Frankreich irrelevant
Country Subentity Code.Code	BBIE	7	für Frankreich irrelevant
Region.Text	BBIE	7	für Frankreich irrelevant
District.Text	BBIE	7	für Frankreich irrelevant
Timezone Offset.Text	BBIE	7	für Frankreich irrelevant
Address Line.Address Line	ASBIE	7	für Frankreich irrelevant
Location Coordinate. Location Coordinate	ASBIE	7	für Frankreich irrelevant

Tab. 11: Irrelevante Elemente für Handel mit französischen Zwischenhändlern über einen Internetshop

Eine Besonderheit in diesem Geschäftsszenario ist die semantische Präzisierung des Adressformats (BBIE `Address Format Code.Code`). Dieses wurde durch einen vorangestellten Qualifier Term semantisch präzisiert, um auszudrücken, dass bei der Angabe der Adresse ein EU-spezifisches Format eingehalten werden soll. Dazu ist zunächst eine Positivdefinition der diesem Geschäftsszenario entsprechenden Konfigurationsparameter als Term *Internetshop France* zu erstellen, da der Qualifier ja gelten soll, wenn diese Parameterausprägungen zutreffen. Im Anschluss erfolgt die Definition einer Bezeichnungsvariation für `Address Format Code.Code`, die den vorangestellten Qualifier Term und die Bezeichnung selber enthält. Um den Bezeichnungsvariation zu aktivieren wird zunächst der Term der Synonymgruppe zugewiesen, und abschließend das entsprechende Synonym der Originalbezeichnung. Eine zusammenfassende Darstellung von Term- und Synonymzuweisungen findet sich in Abb. 45.

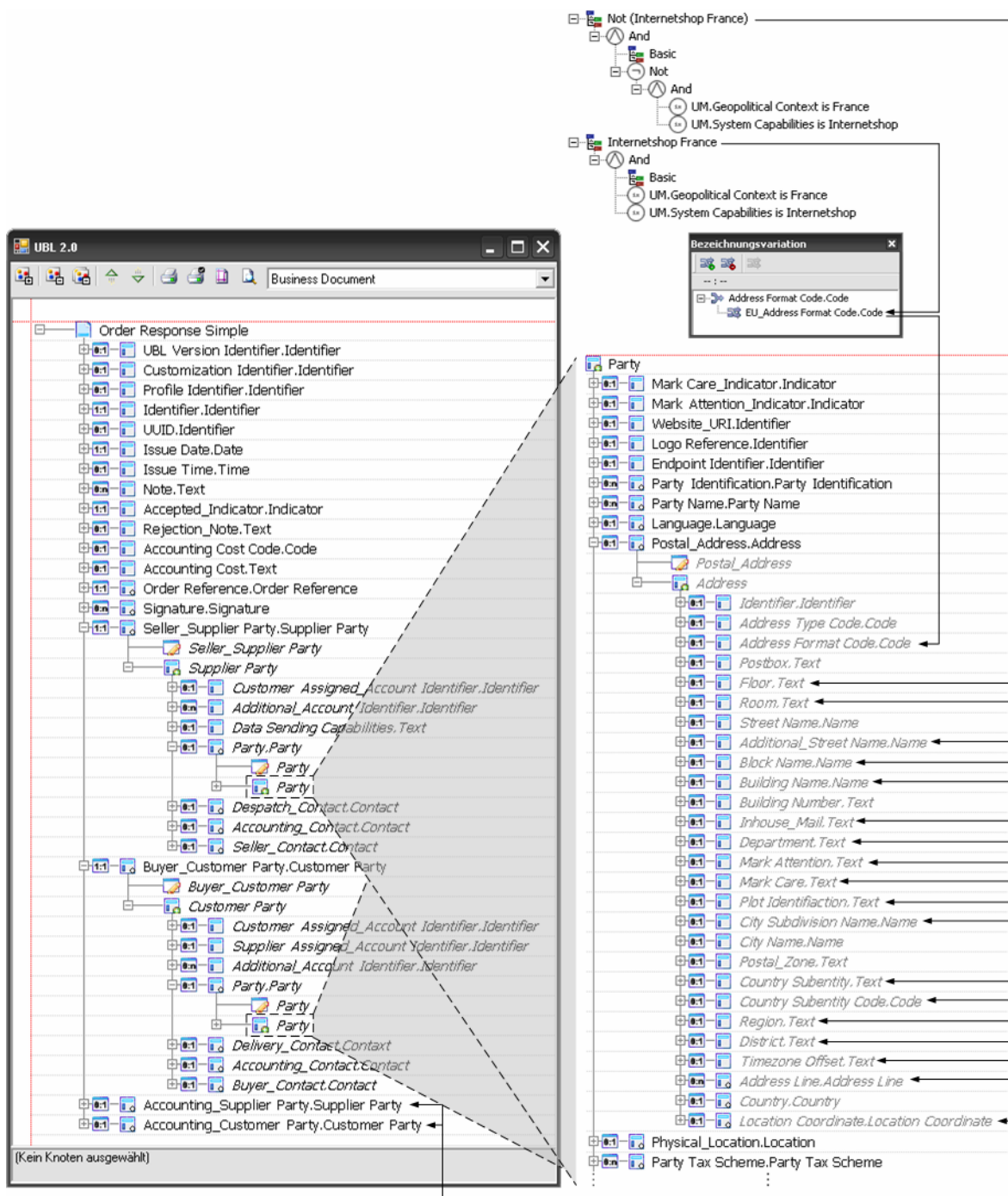


Abb. 45: Zuweisung von Termen für Handel mit französischen Zwischenhändlern über einen Internetshop

Da in der aktuellen Version des H2-Toolsets Elemente nur mit *einem* Term belegt werden können, müssen die bereits für den Handel in Deutschland markierten Elemente modifiziert werden. Dabei wird der ursprüngliche Term wieder entfernt und durch die Konjunktion aus den Termen für Deutschland und Frankreich ersetzt (vgl. Abb. 46).

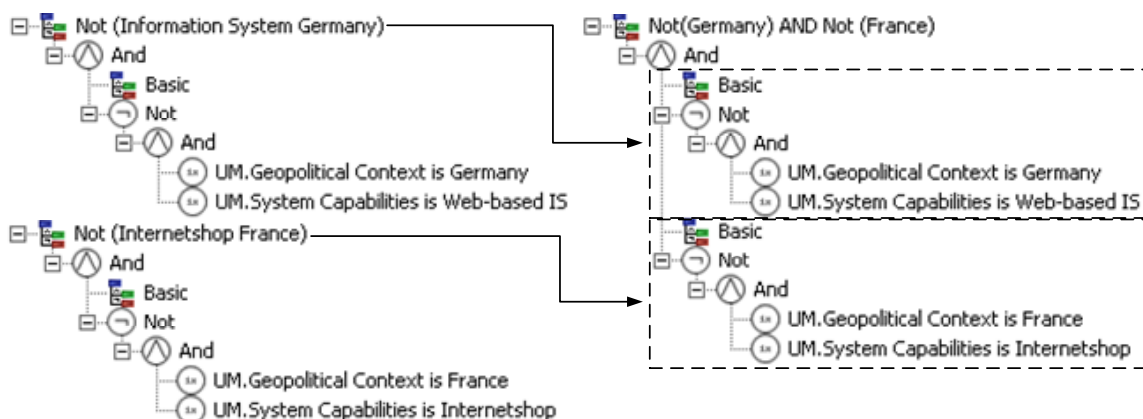


Abb. 46: Kombination der Konfigurationsterme

Direkter Handel mit amerikanischen Einzelhändlern

Der geplante Handel in den USA erfolgt unter Einbeziehung einer Abrechnungsstelle. Im Unterschied zu den vorangehenden Geschäftsszenarien ist daher die Ausblendung der Accounting Parties nicht möglich. Zudem existiert die Besonderheit, dass Anbieter und Kunden aus Ländern kommen, deren Adressstrukturen sich signifikant unterscheiden (bspw. durch die obligatorische Angabe des Bundesstaates durch das BBIE Country Subentity in den USA). Aufgrund der Beschränkung des H2-Toolsets kann daher nur die gemeinsame Obermenge irrelevanter Elemente konfiguriert werden (vgl. Tab. 12).

Name	Typ	Tiefe	Annahme
Floor.Text	BBIE	7	für USA irrelevant
Room.Text	BBIE	7	für USA irrelevant
Additional_Street Name.Name	BBIE	7	für USA irrelevant
Block Name.Name	BBIE	7	für USA irrelevant
Building Name.Name	BBIE	7	für USA irrelevant
Inhouse_Mail.Text	BBIE	7	für USA irrelevant
Department.Text	BBIE	7	für USA irrelevant
Mark Attention.Text	BBIE	7	für USA irrelevant
Mark Care.Text	BBIE	7	für USA irrelevant
Plot Identification.Text	BBIE	7	für USA irrelevant
City Subdivision Name.Name	BBIE	7	für USA irrelevant
Region.Text	BBIE	7	für USA irrelevant
District.Text	BBIE	7	für USA irrelevant
Timezone Offset.Text	BBIE	7	für USA irrelevant
Address Line.Address Line	ASBIE	7	für USA irrelevant
Location Coordinate. Location Coordinate	ASBIE	7	für USA irrelevant

Tab. 12: Irrelevante Elemente für direkten Handel mit amerikanischen Zwischenhändlern

Den Elementen wird der Term *Not (Direct Sale USA)* zugewiesen, um sie für den Kontext als irrelevant zu deklarieren (vgl. Abb. 47). Wie in dem vorangehenden Geschäftsszenario existiert auch hier das Problem, dass die Terme der bereits markierten Elemente so modifiziert werden müssen, dass der resultierende Term der Konjunktion der involvierten Geschäftssituationen entspricht.

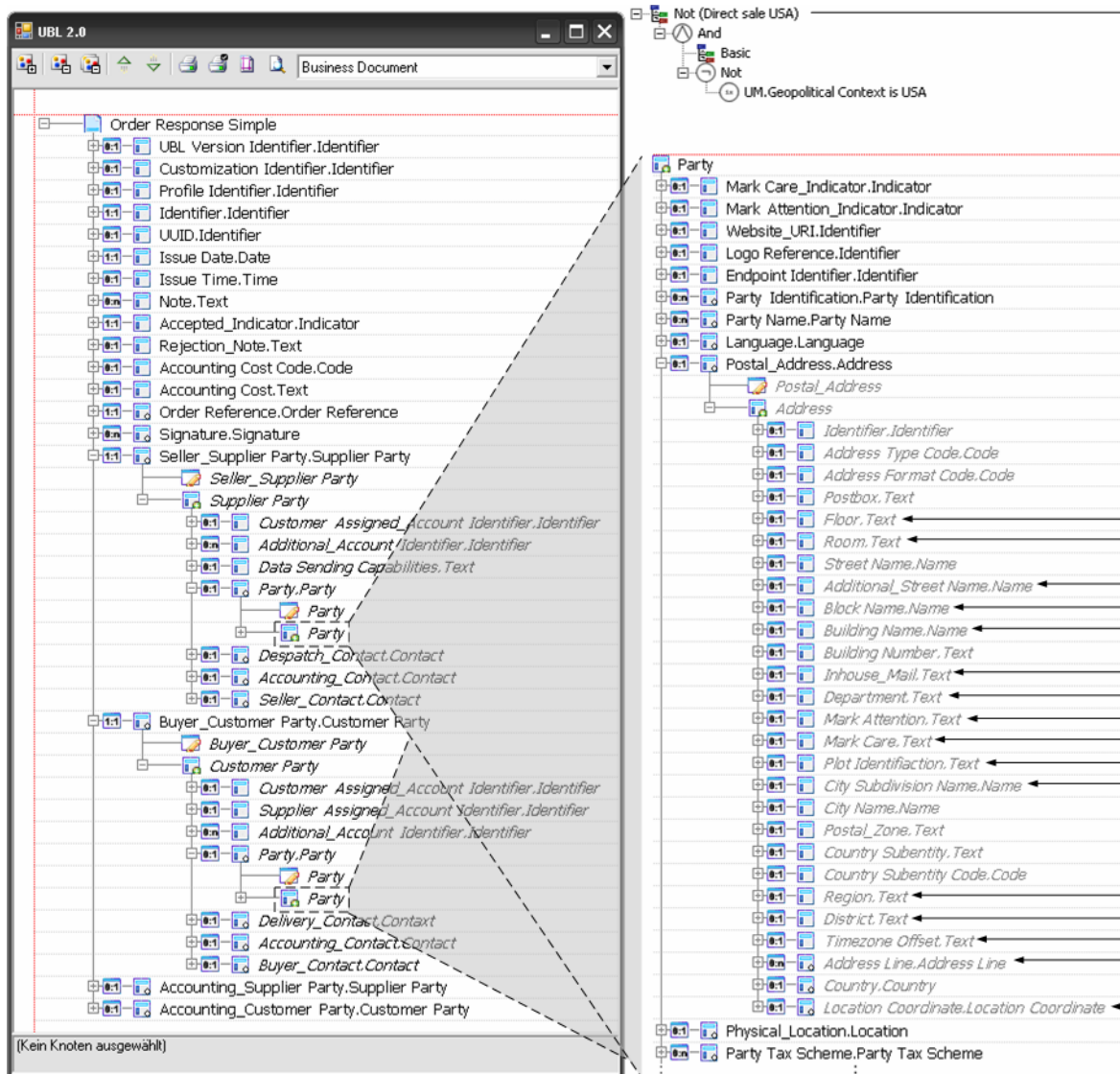


Abb. 47: Zuweisung von Termen für direkten Handel mit amerikanischen Zwischenhändlern

Exkurs: Konfiguration des CCTS-Modells

Neben der in der Fallstudie durchgeführten Konfiguration von Geschäftsdokumenten in UBL ist diese auch für die CCTS exemplarisch zu untersuchen. Dabei ist die Vorgehensweise bei der Zuweisung von Konfigurationstermen zu Elementen des Modells identisch zu dem UBL-Modell. Eine Besonderheit stellt jedoch der Übergang von Core Components

zu Business Information Entities dar. Dieser tritt ein, sobald eine beliebige Konfigurationsparameterausprägung erfüllt ist, da dadurch ein spezifischer Kontext konkretisiert wird. Die Transformation wird im H2-Toolset durch eine Darstellungsvariation der Symbole repräsentiert. Für die Realisierung wird ein neues Image-Set angelegt und mit einem Term belegt, der alle Konfigurationsparameter ODER-verknüpft (vgl. Abb. 48).

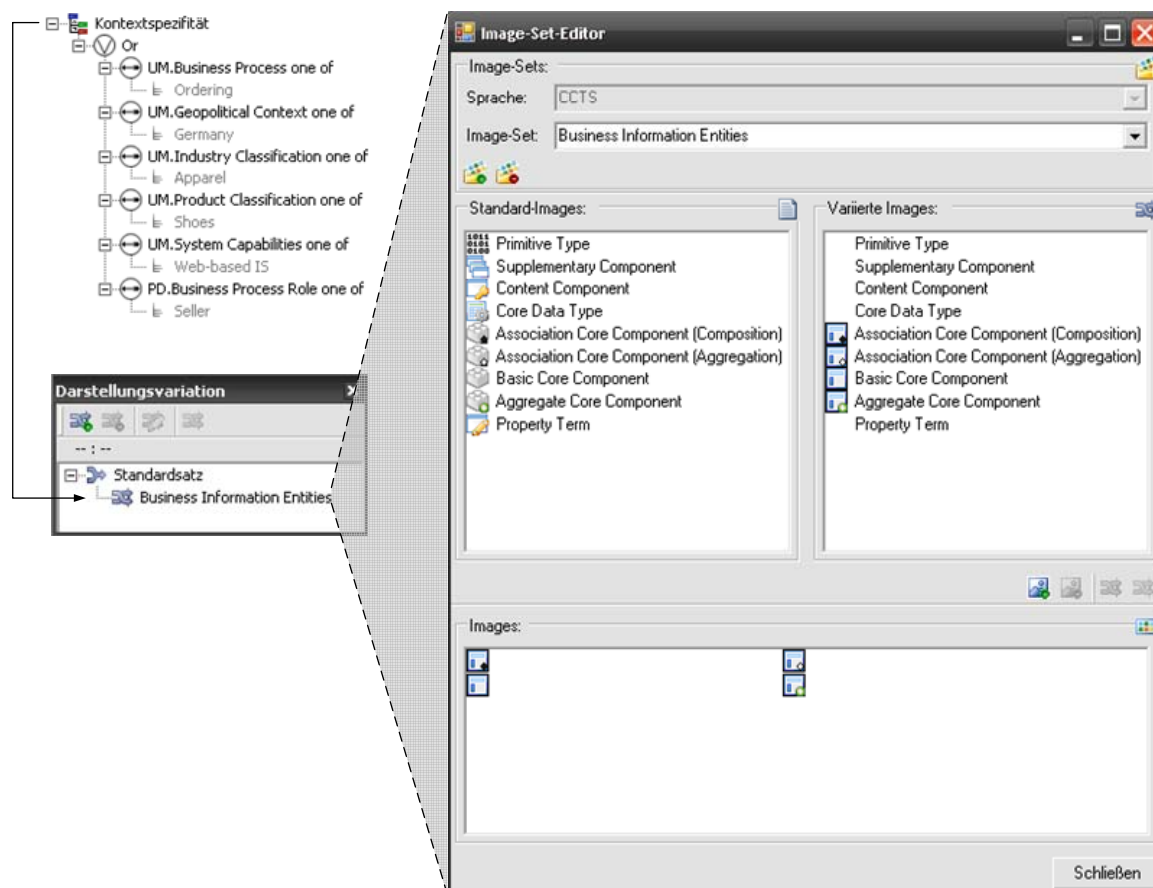


Abb. 48: Darstellungsvariation beim Übergang von Core Components zu Business Information Entities

Um die Transformation von Core Components in Business Information Entities zu vervollständigen, wären die Objekttypen zusätzlich umzubenennen. Wie im vorangegangenen Kapitel bereits ausgeführt unterstützt das H2-Toolset diese Funktion jedoch nicht.

Ergebnis der Konfiguration

Die skizzierte Fallstudie diene als Untersuchungsgrundlage für die Fragestellung, ob das H2-Toolset für die Konfiguration elektronischer Geschäftsdokumente geeignet ist. Dazu ist festzustellen, dass selbst mit den Einschränkungen, denen das H2-Toolset derzeit unterliegt, signifikante Konfigurationsergebnisse erzielbar sind. In Tab. 13 sind die prozentua-

len Reduktionen des Ausgangsdokuments auf oberster Hierarchieebene und auf Ebene des ASBIE Address zusammengefasst:

Order Response Simple	Tiefe 1		Tiefe 7	
	BBIE	ASBIE	BBIE	ASBIE
Grundlegende Konfiguration	- 14,3 %	- 25 %	- 0 %	- 0 %
Deutschland / Informationssystem	- 14,3 %	- 62,5 %	- 66,6 %	- 100 %
Frankreich / Internet	- 14,3 %	- 50 %	- 66,6 %	- 66,6 %
USA / Direkthandel	- 14,3 %	- 25 %	- 58,3 %	- 66,6 %

Tab. 13: Reduktion von Order Response Simple in den einzelnen Konfigurationskontexten

Die Konfiguration der Geschäftsdokumente mit dem H2-Toolset beinhaltet weitere Vorteile. So fungiert das Toolset als einheitliches Repository für die darin abgelegten Geschäftsdokumente und Varianten. Letztere werden nicht separat gespeichert, sondern durch die Formulierung von Konfigurationsbeziehungen über Terme direkt aus dem Ausgangsmodell generiert. Die Varianten sind daher konsistent und weisen keine Redundanzen auf. Ein kombinierter Einsatz des H2-Toolsets mit der CCTS vermeidet das Problem, dass sich nachträgliche Änderungen an Core Components in der ursprünglichen Version der CCTS nicht automatisch auf die Varianten auswirken. Dieser Punkt besitzt insbesondere im Kontext der momentan dynamischen Entwicklung der CCTS Relevanz, da mit dem H2-Toolset die fortdauernde Aktualisierung von Varianten entfällt. Ein weiterer interessanter Aspekt stellt die Tatsache dar, dass UBL-Dokumente durch ihre Abbildung im H2-Toolset syntaxneutral formuliert werden können. Die zentrale Forderung der CCTS, die Semantik von Modellen losgelöst von einer spezifischen Syntax zu betrachten, wird durch die Abbildung im H2-Toolset gewährleistet. Ebenfalls positiv ist zu bewerten, dass sowohl die Modellierung als auch die Konfiguration im H2-Toolset Top-Down erfolgen kann. Hinsichtlich der Modellierung hat dies den Vorteil, dass bedarfsgetriebene Anforderungen bei der Erstellung des Geschäftsdokuments berücksichtigt werden können. Bei der Konfiguration kann die Top-Down-Vorgehensweise den erforderlichen Aufwand erheblich reduzieren, da untergeordnete Elemente gegebenenfalls direkt mit ihrem Väterelement eliminiert werden. Durch die Integration mit der konfigurativen Referenzmodellierung können die Defizite der ursprünglichen Ansätze somit weitgehend vermieden werden.

Bei dem Ergebnis der Konfiguration elektronischer Geschäftsdokumente mit dem H2-Toolset sind auch die Nachteile zu berücksichtigen. Es wird deutlich, dass die Modellierung einer Sprache, der entsprechenden Modelle und insbesondere die Spezifikation der Konfigurationsbeziehung ein enorm aufwändiger Prozess ist. Dieser Aufwand steigt im *worst case* in einem exponentiellen Verhältnis zu der Anzahl der Geschäftsszenarien, da für jede mögliche Kombination dieser Szenarien ein Term zu erstellen ist, der für die Aus-

blendung eines Elements für diese Kombination verantwortlich ist. Auch die nachträgliche Addition eines Geschäftsszenarios in das Repository ist umständlich, da die dort auszublendenden Elemente sukzessive darauf zu untersuchen sind, ob bereits ein Term hinterlegt wurde. Ist dies der Fall, so muss der alte Term entfernt und durch die Konjunktion des neuen Terms und des alten Terms ersetzt werden. Der Einsatz des H2-Toolsets ist somit dahingehend abzuwägen, ob der erzielbare Nutzen den erforderlichen Aufwand für die initiale Modellierung und die ständige Wartung der Termbeziehungen übersteigt.

3.5.2 Evaluation der Grenzen

Das vorangegangene Kapitel verdeutlicht die prinzipielle Eignung der aktuellen Version des H2-Toolsets für die Konfiguration von Referenz-Geschäftsdokumenten. Das Toolset unterliegt jedoch in einigen Bereichen Einschränkungen, die einer vollständigen und effizienten Konfiguration wie in der CCTS bzw. UBL gefordert, entgegenstehen.

Occurrences und Occurrences Explicit

Als besonders problematisch erweist sich die Tatsache, dass bei der Wiederverwendung in anderen Kontexten des H2-Toolsets weder *Occurrences* noch *Occurrences Explicit* jene Eigenschaften aufweisen, die für eine effiziente Modellierung und Konfiguration erforderlich sind. Die Verwendung von *Occurrences* beinhaltet den konzeptionellen Nachteil, dass sie immer auf ihre ursprüngliche Definition referenzieren. Dadurch können Aggregate Business Information Entities, die als Association Business Information Entities in unterschiedlichen Kontexten auftreten, nur einheitlich konfiguriert werden. So enthält z. B. das Geschäftsdokument *Order Response Simple* als ASBIE eine *Supplier Party* und eine *Customer Party*, die beide auf das *ABIE Party* referenzieren. *Party* wiederum enthält spezifische Informationen bzgl. dieser Partei, wie bspw. die Adresse (vgl. Abb. 49). Weisen nun Anbieter und Kunde unterschiedliche Eigenschaften bzgl. der Partei auf, so sind diese nicht separat modellierbar, da der Konfigurationsterm in der aktuellen Version des H2-Toolsets bei der Zuweisung zu einer *Occurrence* nur deren Definition konfiguriert. Eine unterschiedliche Adressstruktur von Anbieter und Kunde, wie sie bspw. durch unterschiedliche Länder auftritt, kann somit nicht abgebildet werden. Die Konfiguration ist daher auf die gemeinsame Obermenge an Eigenschaften beschränkt, die in dem jeweiligen Kontext als relevant bzw. irrelevant eingestuft wird.

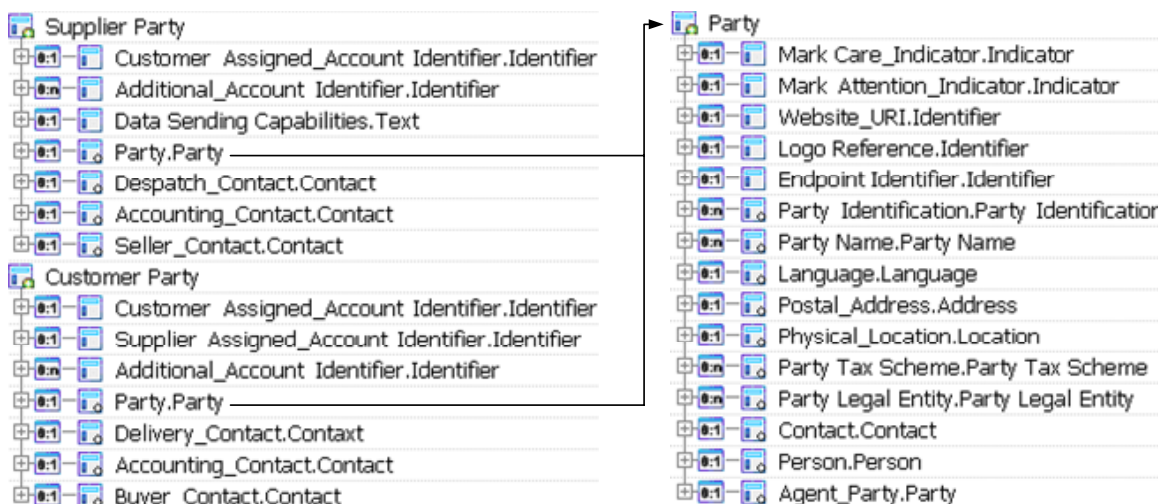


Abb. 49: Referenzen der Supplier und Customer Party

Die Alternative, *Occurrences Explicit* statt *Occurrences* in anderen als den definierenden Kontexten zu verwenden, ist ebenfalls zu verwerfen, da die Modelle nicht vollständig und konsistent erstellbar sind. Dies liegt in der Tatsache begründet, dass *Occurrences Explicit* so konzipiert sind, dass sie bei ihrer Erstellung von der ursprünglichen Definition getrennt werden. Nachträgliche Veränderungen an einer Definition wirken sich daher nicht auf die zugehörige *Occurrence Explicit* aus. Wie aus dem rechten Teil von Abb. 49 ersichtlich wird, referenziert das ABIE *Party* durch das ASBIE *Agent_Party.Party* rekursiv auf sich selbst. Dadurch wird eine bewusste Endlosschleife erzeugt, die mit einer *Occurrence Explicit* nicht abzubilden ist, da für die Erstellung des ABIE *Party* das ASBIE *Agent_Party.Party* erforderlich ist. Wird dieses ASBIE erzeugt, benötigt es als impliziten *Representation Term* wiederum ein ABIE *Party*. Erfolgt dessen Zuweisung als *Occurrence Explicit* unter das ASBIE, so wird es von der ursprünglichen Definition von *Party* getrennt, die noch unvollständig ist, da ihr das ASBIE *Agent_Party.Party* noch nicht zugewiesen wurde. Auch eine Mischform von Beziehungen zwischen ABIE und ASBIE durch *Occurrences Explicit* und *Occurrences* verursacht dieses Problem, da die Verbindung zu der Definition immer getrennt wird, wenn die *Occurrence Explicit* in einem anderen Kontext erstellt wird.

Keine der momentan verfügbaren Alternativen im H2-Toolset stellen eine adäquate Lösung dieses Problems dar, da *im ersten Fall die Modelle konsistent, aber die Konfigurationsmöglichkeiten unzureichend sind*, während der *zweite Fall zwar eine detaillierte Konfiguration ermöglicht, diese aber auf inkonsistenten Modelle stattfindet*. Da keine der Alternativen ein befriedigendes Ergebnis ermöglicht, erscheint es sinnvoll, den Aufwand für eine Erweiterung der jeweiligen Alternative hinsichtlich der gewünschten Funktionalität abzuschätzen.

Die Modellierung mit Occurrences beinhaltet das Problem, dass die den Occurrences zugewiesenen Terme deren Definition referenzieren, und somit eine kontextspezifische Konfiguration nicht stattfinden kann. Um dieses Defizit zu beheben, müsste die starre Kopplung zwischen Konfigurationstermen und Definitionen dynamisiert werden. Hierfür wird eine Erweiterung der Relationen zwischen Termen und Occurrences vorgeschlagen, die occurrencespezifische Annotationen von Konfigurationstermen erlaubt. Die Konfiguration von Definitionen wäre dabei weiterhin möglich, müsste jedoch direkt im definierenden Kontext erfolgen. Als konsistenzsichernde Maßnahmen sollte bei einer Überlagerung zwischen einem Term aus einem definierenden Kontext und jenem aus einem Ausprägungskontext auf einer Objektstruktur in dem Ausprägungskontext die Schnittmenge der Elemente angezeigt werden, die aus der Auswertung beider Terme resultiert. Bei Occurrences Explicit ergibt sich das Problem, dass sie bei ihrer Erstellung von der Definition getrennt werden. Wird die Definition nachträglich modifiziert, so führt dies nicht zu einer Änderung der zugehörigen Occurrence Explicit. Diese Designentscheidung verhindert die Abbildung von Endlosschleifen. Um diese zu ermöglichen müsste die Occurrence Explicit einerseits mit ihrer Definition verbunden bleiben, andererseits sollten Veränderungen der untergeordneten Elemente der Occurrence Explicit möglich sein. Diese wären dynamisch zu speichern und deren Konsistenz gegen mögliche Änderungen an der Definition zu sichern. Die Realisierung dieses Prinzips erscheint ungleich aufwändiger als die erste Alternative, da hier statt einer relativ einfachen Zuweisung von Termen zu Occurrences die Modellstruktur angepasst und überwacht werden müsste. Aus diesem Grund wurde die Konfiguration auf Basis von Occurrences vorgenommen. Dies beinhaltet den weiteren Vorteil, dass die bereits erstellten Modelle hinsichtlich künftiger Erweiterungen kompatibel sind.

Konfigurationsparameter

Ein weiteres Defizit im H2-Toolset ist in der Gestaltung der Konfigurationsparametersprache zu sehen. Wie im vorangegangenen Kapitel bereits erläutert, ist es momentan nicht möglich, selbstdefinierte Konfigurationsparameterkategorien anzulegen. Die daraus resultierende Beschränkung auf Unternehmensmerkmale und Perspektiven bedingt konzeptionelle Probleme bei der Begriffsfassung, da eine teils unsaubere Klassifizierung der Kategorien in Unternehmensmerkmale und Perspektiven zu erfolgen hat. Zudem existieren technische Schwierigkeiten bei der Umsetzung im H2-Toolset. So fordert die CCTS mehrstufige Konfigurationsparameter für gewisse Konfigurationskontextkategorien. Das H2-Toolset ist jedoch nur zweistufig konzipiert und daher bei den Ausprägungen auf eine Hierarchieebene beschränkt. Eine künftige Erweiterung des H2-Toolsets um Hierarchien von Konfigurationsparametern stellt somit eine sinnvolle Ergänzung dar, um die Modellierung und Konfiguration konform zu den Bestimmungen der CCTS zu ermöglichen. Der Implemen-

tierungsaufwand wird dabei als relativ gering eingestuft, da die vorhandene Struktur der Konfigurationsparameter nur um zusätzliche Ebenen erweitert werden müssten. Auch die ontologische Spezifizierung von Konfigurationsparameterbeziehungen im H2-Toolset ist unzureichend. So können bei Bedingungen und Ausschlüssen zwar einzelne Konfigurationsparameterausprägungen angegeben werden, die Annotation von ontologischen Beziehungen an Kategorien dieser Konfigurationsparameter und deren logische Verknüpfung ist derzeit jedoch nicht möglich. Um den Implementierungsaufwand für diese Erweiterung zu reduzieren, könnte auf bereits existierende Prinzipien innerhalb des H2-Toolsets zurückgegriffen werden. So existiert bereits ein ähnliches Konzept für die Erstellung von Konfigurationstermen, die aus *One-of-* bzw. *Is-*Operatoren der Konfigurationsparameter und deren logischen Verknüpfungen zusammengesetzt sind.

Kardinalitäten

In UBL und CCTS besitzen BBIE, ASBIE und Supplementary Components Kardinalitäten. Diese spezifizieren die Häufigkeiten, mit der diese Elemente in spezifischen Kontexten auftreten können. Bei der Adaption von Geschäftsdokumenten in UBL können Kardinalitäten über Schemarestriktionen feingranular angepasst werden. Die Elementselektion im H2-Toolset bietet diese Möglichkeit nicht. Die Auswertung eines Konfigurationsterms zu *false* führt zu einer kompletten Ausblendung des Elements, was einer Kardinalität von (0:0) entspricht. Wertet der Term dagegen zu *true* aus, so führt dies zu einer unveränderten Darstellung des Elements mit der Ausgangskardinalität. Eine detailliertere Behandlung von Kardinalitäten im H2-Toolset wäre daher wünschenswert. Die Implementierung dieses Prinzips stellt interessante Funktionen in Aussicht. So könnten Kardinalitäten bei der Konfiguration automatisch auf ihre Konsistenz überprüft werden. So könnten Modellierer auf Fehler aufmerksam gemacht werden, wenn bspw. ein obligatorisches Element in einer Variante ausgeblendet werden soll. Die Beschränkung auf Projektionen der Ausgangskardinalität, wie sie in UBL vorgesehen ist, erwies sich auch für die Implementierung im H2-Toolset als sinnvolle Restriktion. Eine entsprechende Erweiterung erscheint jedoch relativ komplex, da dieser Aspekt bisher nicht im H2-Toolset berücksichtigt wurde. Dort können momentan zwar Kantentypen innerhalb der Modelle dargestellt werden, ihre Funktion ist jedoch auf die visuelle Repräsentation beschränkt, die keine weitere Funktionalität beinhaltet. Eine umfangreichere Neukonzeption erscheint daher erforderlich.

Bezeichnungsvariation auf Objekttypebene

In der CCTS existieren Core Components und Business Information Entities. Die kontextneutralen CC werden in BIE transformiert, sobald eine Kontextausprägung spezifiziert

wird. Dieser Prozess erfolgt im H2-Toolset durch die Konfiguration des Referenzmodells, bei der eine Auswahl der relevanten Konfigurationsparameter für die Variante erfolgt. Um den Übergang von Core Components in Business Information Entities zu visualisieren, wäre im Spracheditor des H2-Toolsets eine Modifikation der Bezeichnung und des repräsentierenden Symbols erforderlich. Die Änderung des Symbols ist über eine Darstellungsvariation zu realisieren und stellt somit kein Problem dar. Die Modifikation der Bezeichnung dagegen erforderte eine Bezeichnungsvariation auf Objekttypenebene. Dieser Konfigurationsmechanismus existiert in der aktuellen Version des H2-Toolsets jedoch nur für Elemente auf Modellebene. Eine entsprechende Erweiterung für die Sprachebene wäre daher vorteilhaft. Da das grundlegende Konzept bereits existiert, und nur auf eine weitere Ebene zu übertragen ist, erscheint die Realisierung relativ unproblematisch. Implementierungstechnische Details und mögliche Interdependenzen zur Modellebene sind jedoch im Vorfeld zu prüfen.

Stabilität des H2-Toolsets

Eine weitere Grenze der Modellkonfiguration und -adaption mit dem H2-Toolset ist dessen Stabilität und Benutzerfreundlichkeit. Die aktuelle Version, die Möglichkeiten zur Konfiguration bietet, wurde nicht ausreichend auf Anforderungen getestet, die mit der Erstellung von umfangreichen Modellen wie einem Geschäftsdokument verbunden sind. Dadurch ergaben sich während des Modellierungs- und Konfigurationsprozesses teilweise gravierende Stabilitäts- und Ergonomieprobleme. Besonders hervorzuheben ist in diesem Kontext die in Kap. 3.2.2 beschriebene Zuweisung von Attributen zu Objekttypen. Die den Objekttypen annotierten Attribute sind notwendig, um aus den semantischen UBL-Modellen des H2-Toolsets syntaxspezifische Schemadateien zu generieren. Diese Zuweisung konnte aufgrund der Mängel im H2-Toolset nicht vorgenommen werden und musste auf eine konzeptionelle Beschreibung beschränkt bleiben. Weitere Probleme ergaben sich aus dem Umfang der erstellten Modelle. Ein Geschäftsdokument wie `Order Response Simple` umfasst eine große Menge an Definitionen und Ausprägungskopien in anderen Kontexten. Die Performance der Datenbankanbindung erwies sich dabei als problematisch. Die Anweisung, einen Teilbaum zu expandieren, benötigt im H2-Toolset eine gewisse Zeit bis zur Ausführung. Erfolgt in dieser Zeit eine weitere Anweisung, so beendet sich das Programm. Kleinere Fehler wie beim Scrollen in großen Modellen und ausbaufähige Softwareergonomie wie Minimierungs- und Maximierungsfunktionen für Bäume motivieren weiteren Verbesserungsbedarf. Das H2-Toolset sollte daher vor einem offiziellen Release auch mit umfangreicheren Modellen getestet werden, um die Stabilität und Ergonomie zu erhöhen sowie Fehler zu vermeiden.

3.5.3 Evaluation der Potenziale

Bei einer zukünftigen Weiterentwicklung des H2-Toolsets sind insbesondere die zum gegenwärtigen Zeitpunkt noch nicht implementierten Potenziale zu berücksichtigen. Mögliche Erweiterungen können dazu beitragen, den Nutzen der integrierten Konfiguration von elektronischen Geschäftsdokumenten mit dem H2-Toolset zu steigern, und den erforderlichen Aufwand zu reduzieren.

Termmanagement

Im H2-Toolset wird die Relevanz eines Elements für einen Konfigurationsmechanismus durch die Annotation eines Terms ausgedrückt. Das Konfigurationsprinzip über Terme ist somit von entscheidender Bedeutung. Der Vorteil dieses Prinzips ist die Ausdrucksstärke: Durch die logische Verknüpfung der enthaltenen Konfigurationsparameter ist es möglich, jedes durch diese Parameter abzubildende Geschäftsszenario in einem Term zu formulieren. Im Gegenzug steigt jedoch sowohl der Umfang als auch die Anzahl der Terme. Um den damit einhergehenden Konfigurationsaufwand zu reduzieren, ist ein effizientes Termmanagement von essentieller Bedeutung. Ein Ansatz hierfür wäre bspw. die Implementierung eines *Wizards*, der den Modellierer bei der Erstellung relevanter Geschäftsszenarien unterstützt. Auch die Zuweisung der Terme ist verbesserungsfähig: Ist einem bereits mit einem Term belegten Element ein weiterer hinzuzufügen, so ist zunächst der erste Term zu löschen, dieser in einem neu zu erstellenden Term mit dem zweiten zu verknüpfen und die Kombination wieder dem Element hinzuzufügen. Eine bessere Handhabung wäre durch eine dynamische Kombination von Termen zu realisieren, die direkt auf dem Element stattfindet. Die Terme könnten dabei automatisch per Default-Wert oder semiautomatisch durch die Auswahl einer logischen Verknüpfung kombiniert werden. In diesem Zusammenhang wäre auch die Implementierung einer Suchfunktion zu begrüßen, die bereits mit spezifischen Termen belegte Elemente anzeigt. Die Zuweisung von Konfigurationstermen für ähnliche Geschäftsszenarien würde dadurch bedeutend vereinfacht. Eine weitere Möglichkeit stellte die Konzeption eines komplett neuen Konfigurationsprinzips dar. Alternative Strukturen wie *Mindmaps* oder morphologische Kästen zur Auswahl der relevanten Konfigurationsparameter könnten dazu beitragen, die Zuweisung der Konfigurationsmechanismen zu Elementen zu beschleunigen. Vor der Implementierung ist jedoch zu prüfen, ob die Ausdrucksstärke dieser Alternativen den Anforderungen angemessen ist. In diesem Kontext ist auch eine parallele Einführung von Termen und alternativer Konfigurationsprinzipien denkbar, die eine benutzerspezifische Abstufung der Ausdrucksstärke und Benutzerfreundlichkeit unterstützt.

Anwendung generischer Mechanismen

Die Unterstützung generischer Mechanismen kann dazu beitragen, die Qualität der Varianten zu verbessern. In diesem Kontext besitzt insbesondere die *Instanziierung* hohes Anwendungspotenzial. Diese könnte bspw. verwendet werden um Restriktionen an qualifizierte Datentypen zu unterstützen. In der aktuellen Sprache werden diese durch ein Attribut repräsentiert, das als Ausprägungen Tupel aus Restriktionen und Werten enthält. Da die Konfiguration von Attributen im H2-Toolset jedoch nicht möglich ist, sind diese Restriktionen manuell zu definieren. Da die freie Modifikation von Elementen immer die Gefahr inkonsistenter Modellierung beinhaltet, könnte die Instanziierung dazu beitragen, eine generische Restriktion durch eine für den Datentyp vordefinierte Menge an Restriktionen und Werten zu füllen. Die Zuweisung unerlaubter Restriktionen (z. B. *Total Digits* für einen String) könnte dadurch vermieden werden. Auch Qualifier Terms könnten durch die Instanziierung spezifiziert werden. Indem Elemente mehrfach instanziiert werden, könnte auch der Ansatz von STUHEC realisiert werden, der bei der Transformation von Core Components in Business Information Entities mehrere identische BIE mit unterschiedlichen Qualifiern aus einer CC ableitet.¹⁷⁷ Ein Anwendungsbereich der *Aggregation* wäre die in UBL vorgesehene kontextspezifische Erweiterung von Business Information Entities. Die Aggregation könnte dabei durch ein Repository aus Geschäftsdokumentteilen realisiert werden, welches die Erweiterung von Business Information Entities auf Basis eines zu spezifizierenden Regelsatzes ermöglicht. Die Gefahr inkonsistenter Modellierung durch die Erweiterung von Elementen könnte somit zumindest reduziert werden.

Im- und Export von Geschäftsdokumenten

Die Modellierung von Geschäftsdokumenten im H2-Toolset ist sehr zeitintensiv. Um diesen Aufwand zu reduzieren, wäre die XML-Schnittstelle hinsichtlich eines automatischen Imports von XSD-Dateien zu erweitern. Alternativ wäre auch der bereits im H2-Toolset implementierte Excel-Import adaptierbar, um die als Spreadsheets vorliegenden UBL-Geschäftsdokumente als Modelle zu übernehmen. Die Modifikationen sollten dabei sprachbasiert realisiert werden, da die Zuordnung externer Elemente auf die Modellstruktur die Charakteristika der unterliegenden Sprache zu berücksichtigen hat. Einen weiteren interessanten Aspekt stellt die Möglichkeit dar, die im H2-Toolset erstellten Varianten von UBL-Geschäftsdokumenten in XML-Schemata oder andere Syntaxformate zu transformieren. Eine derartige Rücküberführung ermöglichte die mit der konfigurativen Referenzmodellierung realisierten Vorteile in der Praxis.

¹⁷⁷ Vgl. Stuhec (2006a), S. 14 f.

Die Realisierung der aufgeführten Potenziale ermöglichten eine komplette Abdeckung der in der CCTS formulierten Prinzipien und Anforderungen (vgl. Abb. 50). Mit den potenziellen Erweiterungen erwiese sich das H2-Toolset somit auch gegenüber kommerziellen Produkten wie bspw. *GEFEG.FX* als konkurrenzfähig. Die daraus resultierenden Chancen sollten bei der Entscheidung über künftige Entwicklungen berücksichtigt werden.

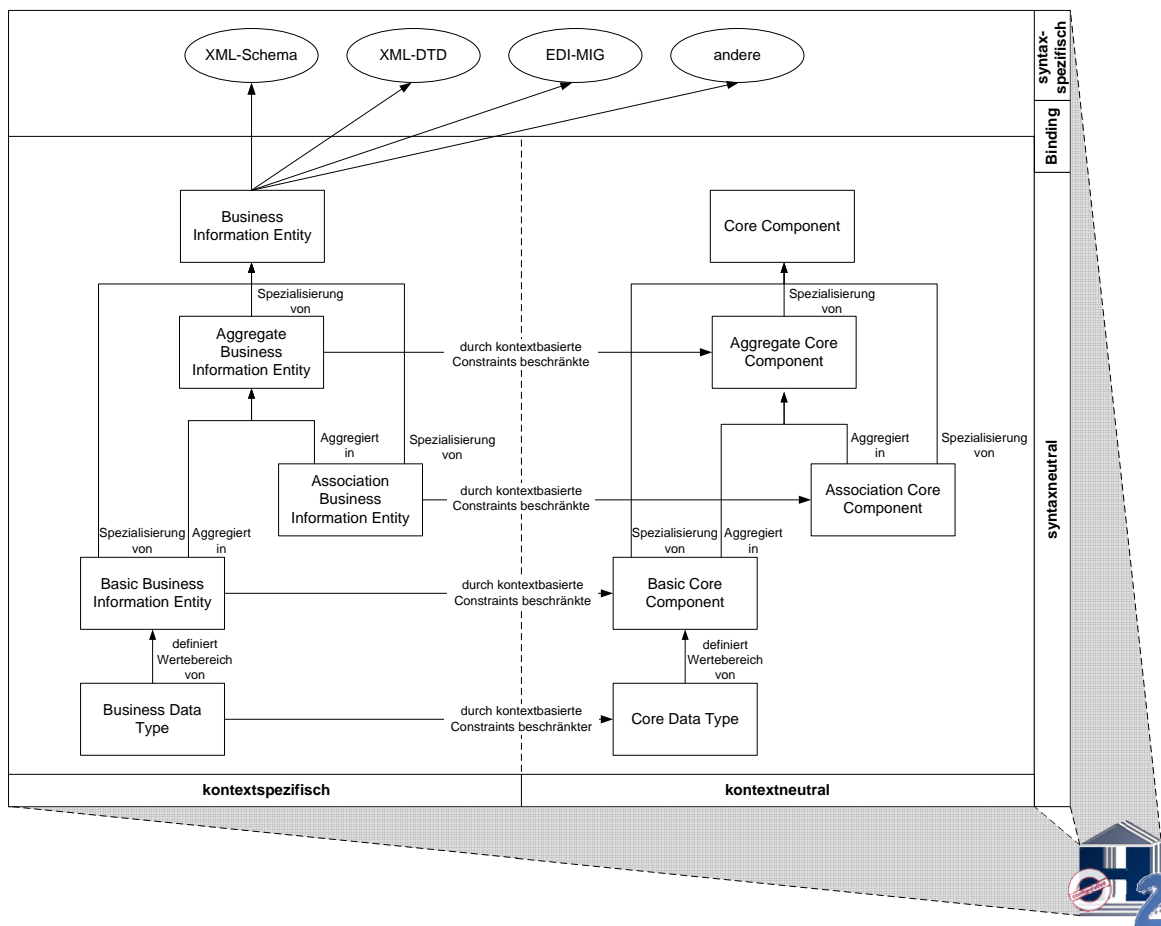


Abb. 50: Abdeckung des CCTS-Konzepts durch das H2-Toolset

4 Fazit und Ausblick

Ausgangspunkt der Konfiguration elektronischer Geschäftsdokumente bildeten die im Rahmen der ursprünglichen Ansätze identifizierten Defizite bei der Variantenerstellung. Die aufbauende Analyse der Vorteilhaftigkeit einer Integration von konfigurativer Referenzmodellierung sowie CCTS bzw. UBL erforderte die Überführung der Fachkonzepte in das H2-Toolset sowie ihre durch die spezifischen Modellierungskriterien der Software bedingte Adaption bzw. Modifikation. Bei der anschließenden Konzeption einer zielsetzungsgerechten Modellierungssprache galt es abzuwägen, welche der den verschiedenen Ansätzen inhärenten Konstrukte explizit als Objekte bzw. Attribute zu modellieren und welche wiederum bereits durch die interne Struktur der Modelle determiniert sind. Die Modellierung des Geschäftsdokuments *Order Response Simple* und der ACC *Address* verdeutlichte den hohen Modellierungsaufwand, den es für die Abbildung eines Referenz-Geschäftsdokuments im H2-Toolset zu bewältigen gilt – die anschließende Konfiguration der Modelle zeigte die Grenzen, die mit der Verwendung des H2-Toolsets und den darin implementierten Mechanismen einhergehen. Ungeachtet dieser Probleme ermöglicht der Einsatz des H2-Toolsets jedoch vielfältige Möglichkeiten der kontextspezifischen Adaption eines Geschäftsdokuments. Darüber hinaus besitzt das H2-Toolset ein enormes Erweiterungspotenzial, um sowohl den Nutzen der Konfiguration zu steigern als auch den erforderlichen Modellierungs- und Konfigurationsaufwand zu reduzieren.

Eine abschließende Bewertung der ursprünglichen Vorgehensweise nach CCTS bzw. UBL sowie der integrierten Anwendung der Ansätze mit dem H2-Toolset ist Tab. 14 zu entnehmen. Die Spalte „H2-Toolset nach Erweiterung“ trägt dabei dem Entwicklungspotenzial des H2-Toolsets Rechnung und spiegelt somit eine Behebung der vorangegangenen dargelegten Probleme sowie entsprechende Erweiterungen der Software wider.

	CCTS	UBL	H2-Toolset aktuell	H2-Toolset nach Erweiterung
Modellierungsansatz	+	-	+	+
Konsistenz und Vermeidung von Redundanzen	~	-	~	+
Erweiterung von Modellvarianten	-	~	~	+
Automatisches Syntaxbinding	-	~	-	+

Tab. 14: Eigenschaften der Ansätze im Überblick

Der *Modellierungsansatz* für die Erstellung individueller Geschäftsdokumente folgt bei der CCTS und im Rahmen des H2-Toolsets eher einem Top-Down-, bei UBL eher einem Bottom-Up-Ansatz. Die *Konsistenz* der Modelle ist bei der CCTS zumindest grundlegend gewährleistet, da die unterliegenden Core Components durch ein Harmonisierungsteam geprüft und normalisiert sind. Probleme können sich jedoch bei Änderung dieser Core Components ergeben, da die zugehörigen Varianten im Repository möglicherweise nicht automatisch aktualisiert werden. Auch werden bei der CCTS Redundanzen durch den vorzunehmenden Abgleich potenzieller Business Information Entities gegen das Repository vermieden. Durch die fehlende Verankerung von Infrastrukturdiensten und die freie Erweiterbarkeit von Varianten besteht bei UBL die Gefahr semantischer Inkonsistenz und redundanter Speicherung. Mit der aktuellen Version des H2-Toolsets existieren zwei konzeptionelle Möglichkeiten, diesen Problemen zu begegnen: Durch die Verwendung von Occurrences können einerseits konsistente Modelle erzeugt werden, deren Varianten jedoch letztlich mehr Elemente enthalten dürfen als tatsächlich erforderlich sind. Die Modellierung mit Hilfe des Occurrence-Explicit-Konstrukts ermöglicht andererseits zwar eine vollständige Konfiguration; die unterliegenden Modelle können jedoch keine Endlosschleifen darstellen und sind deshalb letztlich inkonsistent. Die erweiterte Version des H2-Toolsets vermied dieses Problem. Die *Erweiterung von Modellvarianten* durch Ergänzung von Elementen ist in der CCTS nicht vorgesehen. UBL und die aktuelle Version des H2-Toolsets ermöglichen zwar eine derartige Erweiterung, jedoch erfolgt diese ohne Regelunterstützung. Die erweiterte Version des H2-Toolsets könnte durch den Adaptionismus der Aggregation die Gefahr semantischer Inkonsistenz durch die Implementierung generischer Mechanismen zumindest reduzieren. Ein *automatisches Syntaxbinding* existiert in der CCTS nicht, da diese ein implementierungsunabhängiger Metastandard ist. Auch die aktuelle Version des H2-Toolsets verfügt nicht über eine derartige Funktion. Bei UBL existiert zumindest ein implizites Syntaxbinding, da deren Geschäftsdokumente auf XML-Schemata basieren. Die erweiterte Version des H2-Toolsets könnte bei entsprechender Erweiterung beliebig viele Syntaxformate unterstützen.

Es bleibt somit festzuhalten, dass die Integration von konfigurativer Referenzmodellierung und elektronischen Geschäftsdokumenten im H2-Toolset die Defizite der ursprünglichen Ansätze weitgehend eliminieren kann. Die Entscheidung für einen entsprechenden Einsatz sollte sich jedoch an der Frage orientieren, ob der resultierende Nutzen den Aufwand für Modellerstellung und Wartung der Konfigurationsbeziehungen übersteigt. Die in der Arbeit untersuchten Grenzen und Erweiterungspotenziale abstrahieren von implementierungstechnischen Details. Zukünftiger Forschungsbedarf besteht daher in der theoretischen und praktischen Ausgestaltung dieser Aspekte. Auch die relativ starre Festlegung der konfigurativen Referenzmodellierung auf Unternehmensmerkmale und Perspektiven bedarf bei Übertragung auf andere Anwendungsdomänen einer theoretischen Neukonzeption.

Literaturverzeichnis

- Alpar, P.; Grob, H. L.; Weimann, P.; Winter, R.: Anwendungsorientierte Wirtschaftsinformatik: Strategische Planung, Entwicklung und Nutzung von Informations- und Kommunikationssystemen. Braunschweig 2002.
- Badakhchani, H.: Introduction to RosettaNet (2004). <http://dev2dev.bea.com/pub/a/2004/12/RosettaNet.html>. Abrufdatum: 19.12.2006.
- Becker, J.; Delfmann, P.; Knackstedt, R.: Adaption fachkonzeptioneller Referenzprozessmodelle. In: *Industriemanagement*, 20 (2004a), S. 19-22.
- Becker, J.; Delfmann, P.; Knackstedt, R.: Konstruktion von Referenzmodellierungssprachen: Ein Ordnungsrahmen zur Spezifikation von Adaptionsmechanismen für Informationsmodelle. In: *Wirtschaftsinformatik*, 46 (2004b), S. 251-264.
- Becker, J.; Delfmann, P.; Knackstedt, R.; Kuropka, D.: Konfigurative Referenzmodellierung. In: *Wissensmanagement mit Referenzmodellen. Konzepte für die Anwendungssystem- und Organisationsgestaltung*. Hrsg.: Becker, J., Knackstedt, R., Heidelberg 2002, S. 25-144.
- Becker, J.; Delfmann, P.; Rieke, T.: RefMod06 - Wiederverwendung fachkonzeptioneller Softwaremodelle für kleine und mittlere Softwareunternehmen durch adaptive, komponentenorientierte Referenzmodellierung. *Proceedings of the Eröffnungskonferenz Software Engineering*, Berlin 2004.
- Becker, J.; Janiesch, C.; Knackstedt, R.; Kramer, S.; Seidel, S.: Konfigurative Referenzmodellierung mit dem H2-Toolset. In: *Arbeitsberichte des Instituts für Wirtschaftsinformatik Nr. 114*. Münster 2006a.
- Becker, J.; Janiesch, C.; Knackstedt, R.; Müller-Wienbergen, F.; Seidel, S.: H2 for Reporting - Analyse, Konzeption und kontinuierliches Metadatenmanagement von Management-Informationssystemen. In: *Arbeitsberichte des Instituts für Wirtschaftsinformatik Nr. 115*. Münster 2007.
- Becker, J.; Janiesch, C.; Pfeiffer, D.: Towards more Reuse in Conceptual Modeling: A Combined Approach using Contexts. *Proceedings of the 19th International Conference on Advanced Information Systems Engineering (CAiSE 2007) Forum*, Trondheim 2007.
- Becker, J.; Janiesch, C.; Seidel, S.; Brelage, C.: A Framework for Situational and Evolutionary Language Adaptation in Information Systems Development. *Proceedings of the 15th International Conference on Information Systems Development (ISD 2006)*, Budapest, Hungary 2006b.
- Becker, J.; Janiesch, C.; Seidel, S.; Brelage, C.; Crisandt, J.: Specifying Modeling Languages with H2. *Proceedings of the 1st Workshop on Meta-Modelling and Corresponding Tools (WoMM 2005)*, Essen 2005.
- Becker, J.; Knackstedt, R.: Konstruktion und Anwendung fachkonzeptioneller Referenzmodelle im Data Warehousing. Heidelberg 2003.
- Becker, J.; Knackstedt, R.: Referenzmodellierung im Data-Warehousing: State-of-the-Art und konfigurative Ansätze für die Fachkonzeption. In: *Wirtschaftsinformatik*, 46 (2004), S. 39-49.
- Becker, J.; Niehaves, B.; Knackstedt, R.: Bezugsrahmen zur epistemologischen Positionierung der Referenzmodellierung. In: *Referenzmodellierung: Grundlagen, Techniken und domänenbezogene Anwendung*. Hrsg.: Becker, J., Delfmann, P., Heidelberg 2004, S. 1-18.
- Bosak, J.: XML, Java, and the future of the Web (1997). <http://www.ibiblio.org/pub/sun-info/standards/xml/why/xmlapps.html>. Abrufdatum: 19.12.2006.

- Bray, T.; Paoli, J.; Sperberg-McQuenn, C. M.; Maler, E.; Yergeau, F.: Extensible Markup Language (XML) 1.0 (2006). <http://www.w3.org/TR/2006/REC-xml-20060816/>.
Abrufdatum: 19.12.2006.
- Chen, P. P.: The Entity-Relationship Model - Toward a Unified View on Data. In: ACM Transactions on Database-Systems, 1 (1976), S. 9-36.
- Clark, J.: Comparison of SGML and XML (1997). <http://www.w3.org/TR/NOTE-sgml-xml-971215.html>. Abrufdatum: 19.12.2006.
- Crawford, M.: Universal Business Language: Realizing eBusiness XML (2002).
xml.coverpages.org/Crawford-UBL200212.pdf Abrufdatum: 10.01.2007.
- Damodaran, S.: B2B integration over the internet with XML: RosettaNet successes and challenges. Proceedings of the WWW 2004, New York 2004.
- Delfmann, P.: Adaptive Referenzmodellierung: Methodische Konzepte zur Konstruktion und Anwendung wiederverwendungsorientierter Informationsmodelle. Dissertation, Münster. Münster 2006.
- ebXML: ebXML Whitepapers: Enabling Electronic Business with ebXML (2000).
http://www.ebxml.org/white_papers/whitepaper.htm. Abrufdatum: 22.12.2006.
- ebXML: Catalogue of Common Business Processes v1.0 (2001a). <http://www.ebxml.org/specs/bpPROC.pdf>. Abrufdatum: 02.01.2007.
- ebXML: Catalogue of Context Drivers v1.04 (2001b). <http://www.ebxml.org/specs/ccDRIV.pdf>. Abrufdatum: 05.01.2007.
- ebXML: ebXML Core Components Structure v1.04 (2001c). <http://www.ebxml.org/specs/ccSTRUCT.pdf>. Abrufdatum: 29.12.2006.
- ebXML: Naming Conventions for Core Components (2001d). <http://www.ebxml.org/specs/ebCCNAM.pdf>. Abrufdatum: 29.12.2006.
- ebXML: About ebXML (2006a). <http://www.ebxml.org/geninfo.htm> Abrufdatum: 12.12.2006.
- ebXML: ebXML Industry Support (2006b). <http://www.ebxml.org/endorsements.htm>. Abrufdatum: 22.12.2006.
- Emmelhainz, M. A.: Electronic Data Interchange - A Total Management Guide. New York 1990.
- Fettke, P.; Loos, P.: Perspectives on Reference Modeling. In: Reference Modeling for Business System Analysis. Hrsg.: Fettke, P., Loos, P., Hershey 2007, S. 1-20.
- Glushko, R. J.; McGrath, T.: Document Engineering: Analyzing and Designing Documents for Business Informatics and Web Services. London 2005.
- Harold, E.-R.; Means, W.-S.: XML in a nutshell. Köln 2005.
- ISO/IEC: Information technology — Metadata registries (MDR) — Part 5: Naming and identification principles (2005). [http://standards.iso.org/ittf/PubliclyAvailableStandards/c035347_ISO_IEC_11179-5_2005\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/c035347_ISO_IEC_11179-5_2005(E).zip).
- Janiesch, C.; Thomas, S. M.: Business Document Taxonomy: Comparison of the State-of-the-art and Recommendations for Future Applications. In: Interoperability in Business Information Systems, 2 (2006) 2, S. 59 - 78.
- Janner, T.; Schmidt, A.; Schroth, C.; Stuhec, G.: From EDI to UN/CEFACT: An Evolutionary Path Towards a Next Generation e-Business Framework. 2006.
- Jauhiainen, S.; Lehtonen, O.; Ranta-aho, P.-P.; Rogemond, N.: B2B Integration: past, present, and future (2005). <http://www.soberit.hut.fi/T-86/T-86.161/2005/B2Bi-final.pdf>. Abrufdatum: 22.12.2006.
- Klein, S.: Interorganisationssysteme und Unternehmensnetzwerke: Wechselwirkungen zwischen organisatorischer und informationstechnischer Entwicklung. Zugleich Habilitationsschrift im Fach Betriebswirtschaftslehre an der Universität St. Gallen, 1994. Wiesbaden 1998.

- Knackstedt, R.: Fachkonzeptionelle Referenzmodellierung einer Managementunterstützung mit quantitativen und qualitativen Daten: Methodische Konzepte zur Konstruktion und Anwendung. Zugleich Dissertation Münster. Berlin 2006.
- Lamprecht, A.: Elektronischer Datenaustausch (EDI) in Verbundgruppen. Wiesbaden 1998.
- Maler, E.: XML for e-Business (2003). <http://xml.coverpages.org/Maler-CSW-xml-for-ebusiness.pdf>. Abrufdatum: 10.01.2007.
- Mehnen, H.: Telekommunikation in Handel, Transport und Verwaltung nach internationalem Standard: ISO-Standard 9735 EDIFACT. Eschborn 1989.
- Mittelstraß, J.: Enzyklopädie Philosophie und Wissenschaftstheorie. Stuttgart 1995.
- Neuburger, R.: Electronic Data Interchange: Einsatzmöglichkeiten und ökonomische Auswirkungen. Wiesbaden 1994.
- OASIS: Guidelines for the Customization of UBL v1.0 Schemas - Working Draft 1.0-beta 3 (2004). <http://docs.oasis-open.org/ubl/cd-UBL-1.0/doc/cm/wd-ubl-cm-sc-cmguidelines-1.0.html>. Abrufdatum: 11.01.2007.
- OASIS: Universal Business Language v2.0 Standard (2006). <http://docs.oasis-open.org/ubl/os-UBL-2.0/UBL-2.0.html>. Abrufdatum: 09.01.2007.
- OASIS: OASIS Universal Business Language FAQ (2007). <http://www.oasis-open.org/committees/ubl/faq.php>. Abrufdatum: 08.01.2007.
- OASIS ebXML Joint-Committee: The Framework for eBusiness (2006). <http://www.oasis-open.org/committees/download.php/17817/ebxmljc-WhitePaper-wd-r02-en.pdf>. Abrufdatum: 24.12.2006.
- OASIS UBL TC: UBL Common Library v2.0 (2007). <http://docs.oasis-open.org/ubl/os-UBL-2.0/mod/common/UBL-CommonLibrary-2.0.xls>. Abrufdatum: 10.01.2007.
- Pfeiffer, H. K. C.: The diffusion of Electronic Data Interchange. Heidelberg 1992.
- Picot, A.; Neuburger, R.; Niggel, J.: Ökonomische Perspektiven eines Electronic Data Interchange. In: Information Management, 2 (1991), S. 22-29.
- Ray, E.-T.: Einführung in XML. Köln 2001.
- RosettaNet: RosettaNet Implementation Framework: Core Specification (2001). <http://xml.coverpages.org/RNIF-Spec020000.pdf>. Abrufdatum: 20.01.2007.
- RosettaNet: About RosettaNet Standards (2006a). <http://portal.rosettanet.org/cms/sites/RosettaNet/Standards/index.html>. Abrufdatum: 21.12.2006.
- RosettaNet: About RosettaNet: Who we are (2006b). <http://portal.rosettanet.org/cms/sites/RosettaNet/About/Who/index.html>. Abrufdatum: 21.12.2006.
- RosettaNet: PIPs (2006c). <http://portal.rosettanet.org/cms/sites/RosettaNet/Standards/RStandards/pip/index.html>. Abrufdatum: 21.12.2006.
- RosettaNet: RosettaNet Dictionaries (2006d). <http://portal.rosettanet.org/cms/sites/RosettaNet/Standards/RStandards/dictionary/index.html>. Abrufdatum: 21.12.2006.
- RosettaNet: RosettaNet Implementation Framework (2006e). <http://portal.rosettanet.org/cms/sites/RosettaNet/Standards/RStandards/rnif/index.html>. Abrufdatum: 21.12.2006.
- RosettaNet: RosettaNet Standards (2006f). <http://portal.rosettanet.org/cms/sites/RosettaNet/Standards/RStandards/index.html>. Abrufdatum: 21.12.2006.
- RosettaNet: RosettaNet Standards Methodology (2006g). <http://portal.rosettanet.org/cms/sites/RosettaNet/Standards/Methodology/index.html>. Abrufdatum: 21.12.2006.
- Scheer, A.-W.: Architektur integrierter Informationssysteme: Grundlagen der Unternehmensmodellierung. Berlin 1992.
- Scheer, A.-W.: Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse. Berlin, Heidelberg, New York 1997.

- Schulze, D.: Grundlagen der wissensbasierten Konstruktion von Modellen betrieblicher Systeme. Aachen 2001.
- Schütte, R.: Grundsätze ordnungsmäßiger Referenzmodellierung: Konstruktion konfigurations- und anpassungsorientierter Modelle. Wiesbaden 1998.
- Schwegmann, A.: Objektorientierte Referenzmodellierung - Theoretische Grundlagen und praktische Anwendung. Wiesbaden 1999.
- Sedran, T.: Wettbewerbsvorteile durch EDI? In: Information Management, 2 (1991), S. 16-21.
- Stuhec, G.: How to Solve the Business Standards Dilemma: The Context Driven Business Exchange (2005). <https://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/docs/library/uuid/a6c5dce6-0701-0010-45b9-f6ca8c0c6474>. Abrufdatum: 27.12.2006.
- Stuhec, G.: How to Solve the Business Standards Dilemma: CCTS Key Model Concepts (2006a). <https://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/docs/library/uuid/1b873fc3-0901-0010-f7bc-9518e1aed0cf>. Abrufdatum: 29.12.2006.
- Stuhec, G.: How to Solve the Business Standards Dilemma: The CCTS based Core Data Types (2006b). <https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/500db5c9-0e01-0010-81aa-d73cdd30df9a>. Abrufdatum: 31.12.2006.
- Stuhec, G.; Crawford, M.: How to Solve the Business Standards Dilemma: The CCTS Standards Stack (2006). <https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/30d35ece-5c67-2910-64aa-cb331726ee1c>. Abrufdatum: 08.01.2007.
- Taschek, J.: Interview with Jon Bosak: UBL raising E-Com Hopes (2003). eWeek.
- Teubner, R. A.: Organisations- und Informationssystemgestaltung: Theoretische Grundlagen und integrierte Methoden. Dissertation, Münster. Wiesbaden 1999.
- Thomas, H. E.: Die neuen EDI-Wege: Ein Überblick. Berlin, Wien, Zürich, Beuth 1999.
- UBL Marketing Subcommittee: UBL: The Next Step for Global E-Commerce (2002). <http://www.oasis-open.org/committees/ubl/msc/200204/ubl.pdf>. Abrufdatum: 09.01.2007.
- UN/CEFACT: Core Components Technical Specification: Part 8 of the ebXML Framework (2003a). http://www.unece.org/cefact/ebxml/CCTS_V2-01_Final.pdf. Abrufdatum: 12.12.2006.
- UN/CEFACT: UN/CEFACT Modeling Methodology (UMM) User Guide (2003b). http://www.unece.org/cefact/umm/UMM_userguide_220606.pdf. Abrufdatum: 07.01.2007.
- UN/CEFACT: Core Component Library (2006a). <http://www.unece.org/cefact/codesfortrade/unccl/CCL06A.xls>. Abrufdatum: 07.01.2007.
- UN/CEFACT: Core Components Technical Specification 2nd Edition (Working Draft / Draft B) (2006b). http://www.untmg.org/index.php?option=com_docman&task=docclick&Itemid=137&bid=43&limitstart=0&limit=5. Abrufdatum: 26.12.2006.
- UN/CEFACT: Core Components Technical Specification Version 2.2 (Draft Version) - Appendix A: Core Data Types (2006c). http://www.untmg.org/index.php?option=com_docman&task=docclick&Itemid=137&bid=42&limitstart=0&limit=5. Abrufdatum: 31.12.2006.
- UN/CEFACT: UML Profile for Core Components (BCSS) - Candidate for Version 1.0 - Working Draft for public review (2006d). http://www.untmg.org/index.php?option=com_docman&task=docclick&Itemid=137&bid=34&limitstart=0&limit=5. Abrufdatum: 08.01.2007.

- UN/CEFACT; OASIS: Context and Re-Usability of Core Components: ebXML Core Components (2001). www.ebxml.org/specs/ebCNTXT.pdf. Abrufdatum: 22.12.2006.
- vom Brocke, J.: Referenzmodellierung: Gestaltung und Verteilung von Konstruktionsprozessen. Dissertation, Münster. Münster 2003.
- vom Brocke, J.; Buddendick, C.: Konstruktionstechniken für die Referenzmodellierung: Systematisierung, Sprachgestaltung und Werkzeugunterstützung. In: Referenzmodellierung: Grundlagen, Techniken und domänenbezogene Anwendung. Hrsg.: Becker, J., Delfmann, P., Heidelberg 2004a, S. 19-50.
- vom Brocke, J.; Buddendick, C.: Organisationsformen in der Referenzmodellierung - Forschungsbedarf und Gestaltungsempfehlungen auf Basis der Transaktionskostentheorie. In: Wirtschaftsinformatik, 46 (2004b) 5, S. 341 - 352.
- W3C: XML Schema (2004). <http://www.w3.org/XML/Schema>. Abrufdatum: 11.01.2007.
- Weitzel, T.; Harder, T.; Buxmann, P.: Electronic Business und EDI mit XML. Heidelberg 2001.
- Wittenbrink, H.; Köhler, W.; Bergmann, O.; Jung, B.; Witt, A.; Sasaki, F.; Lenz, E.-A.; Trippel, T.; Milde, J.-T.; Poenninghaus, J.: XML - Wissen, das sich auszahlt. Berlin 2003.
- xCBL: xCBL Home (2007). <http://www.xcbl.org/>. Abrufdatum: 09.01.2007.

Anhang

A ebXML-Framework

ebXML ist eine im Jahr 1999 von der UN/CEFACT und OASIS gegründete Initiative, die durch die Entwicklung einer breiten Palette verschiedener Standards den elektronischen Handel von Unternehmen unterstützt.¹⁷⁸ Die Vision von ebXML ist die Schaffung eines einzelnen, großen elektronischen Marktplatzes, auf dem sich Unternehmen jeder Größe und jeder geographischen Region treffen, um mittels XML basierter Nachrichten elektronischen Handel zu betreiben.¹⁷⁹ Dazu wird ein kompletter Satz von Spezifikationen bereitgestellt, der auf etablierten sowie offenen Standards wie *TCP/IP (Transmission Control Protocol / Internet Protocol)*, HTTP und XML basiert. Ein besonderer Fokus von ebXML liegt dabei auf elektronischer Interoperabilität. Diese wird durch die Bereitstellung eines offenen, plattformunabhängigen und implementierungsneutralen semantischen Frameworks, das Unternehmen während des gesamten Handelsprozesses unterstützt, gefördert.¹⁸⁰ Das Framework unterteilt sich in drei Komponenten:¹⁸¹

- Eine Infrastruktur, die die Interoperabilität des Datenaustauschs sicherstellt. Diese ist in zwei weitere Komponenten untergliedert,
 - einen Mechanismus für den Nachrichtentransport, inklusive definierter Schnittstellen, Regeln für die Kapselung und einem Versende- und Sicherheitsmodell.
 - einem *Business Service Interface*, das ein- und ausgehende Nachrichten weiterverarbeitet.
- Ein Rahmen, der die semantische Interoperabilität gewährleistet. Dies erfolgt durch
 - ein Metamodell für die Definition von Geschäftsprozessen und Informationsmodellen.
 - eine Menge wiederverwendbarer Geschäftslogik auf der Basis von Core Components und XML-Vokabularen.
 - einen Prozess zur Definition von Nachrichtenstrukturen aus den Aktivitäten des Geschäftsprozessmodells.

¹⁷⁸ Vgl. ebXML (2006a).

¹⁷⁹ Vgl. ebXML (2000).

¹⁸⁰ Vgl. ebXML (2000).

¹⁸¹ Vgl. ebXML (2000).

- Eine Methode, um den gesamten Handelsprozess der Unternehmen zu unterstützen, d. h. von der Suche über die Vertragsanbahnung bis zur Abwicklung. Dies wird realisiert durch
 - ein gemeinsames Repository, bei dem sich Unternehmen registrieren und sie Geschäftsprozesse anderer Unternehmen identifizieren können.
 - einen Prozess für die Definition und Zustimmung zu einem formalen *Collaboration Protocol Agreement*.
 - ein gemeinsames Repository für Unternehmensprofile, Geschäftsprozessmodelle und zugehörige Nachrichtenstrukturen.

Der Geschäftsprozess zwischen Unternehmen nach ebXML ist folgender Abbildung zu entnehmen (vgl. Abb. 51):

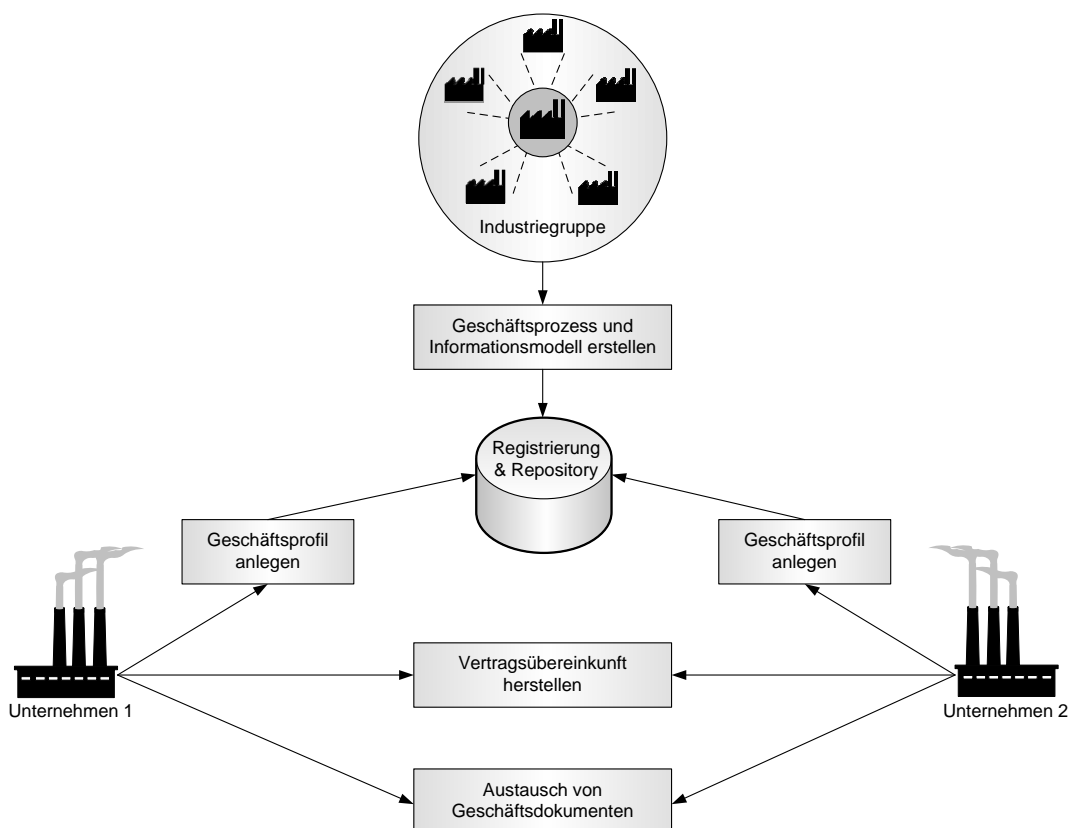


Abb. 51: Durch die ebXML-Spezifikationen unterstützte Aktivitäten¹⁸²

Die ebXML-Standards sind anhand der unterstützten Aktivitäten ausgerichtet. Innerhalb des oben skizzierten Frameworks existieren:¹⁸³

¹⁸² In Anlehnung an OASIS ebXML Joint-Committee (2006), S. 5.

- Die *Core Components Technical Specification* für den Austausch eindeutiger Geschäftsdaten in einem standardisierten Format.
- Der *Registry-Dienst* und die *Registry-Informationsmodelle* für die Registrierung und Bereitstellung von eBusiness-Artifakten und –diensten.
- Das *Collaboration Protocol Profile* und *-Agreement* für die Berücksichtigung technischer Aspekte beim Datenaustausch zwischen Unternehmen.
- Der *Nachrichtendienst* für die Bereitstellung eines sicheren und zuverlässigen Transports.
- Das *Business Process Specification Schema (BPSS)* für die Definition der Geschäftsprozesse.

¹⁸³ Vgl. OASIS ebXML Joint-Committee (2006), S. 5.

B Übernahme potenzieller ABIE in das Repository

Die *UN/CEFACT Modeling Methodology (UMM)* ist eine Modellierungsmethode, um Geschäftsprozesse der Nutzer zu analysieren und relevante Kontextausprägungen sowie potenzielle Aggregate Business Information Entities zu identifizieren. Potenzielle ABIE werden innerhalb des Repositories auf Ähnlichkeiten zu bereits vorhandenen ABIE verglichen und diesem gegebenenfalls hinzugefügt. Den Vergleichsprozess zwischen dem potenziellen ABIE und dem Repository visualisiert die EPK in Abb. 52: Liegt eine exakte Übereinstimmung zwischen den beiden ABIE vor, so wird nur der neue Verwendungskontext registriert und übermittelt. Sind die ABIE nicht identisch, jedoch strukturell ähnlich, so kann eine Modifikation des ABIE innerhalb des Repositories beantragt und übermittelt werden. Liegt keine Ähnlichkeit zwischen dem potenziellen ABIE und den ABIE innerhalb des Repositories vor, so ist zu prüfen, ob das potenzielle ABIE aus einer Aggregate Core Component abgeleitet werden kann. Ist die Ableitung aus einer ACC möglich, so wird die Relation zwischen ABIE und ACC sowie der Verwendungskontext registriert und übermittelt. Sind das ABIE und die ACC strukturell ähnlich, so wird eine Anpassung der ACC sowie der Verwendungskontext beantragt und übermittelt. Sind das ABIE und die vorhandenen ACC nicht ähnlich, so wird eine neue ACC inklusive dem abgeleiteten ABIE und dem Verwendungskontext beantragt. Die übermittelten Änderungen werden von einer Arbeitsgruppe geprüft und mit dem Repository harmonisiert.

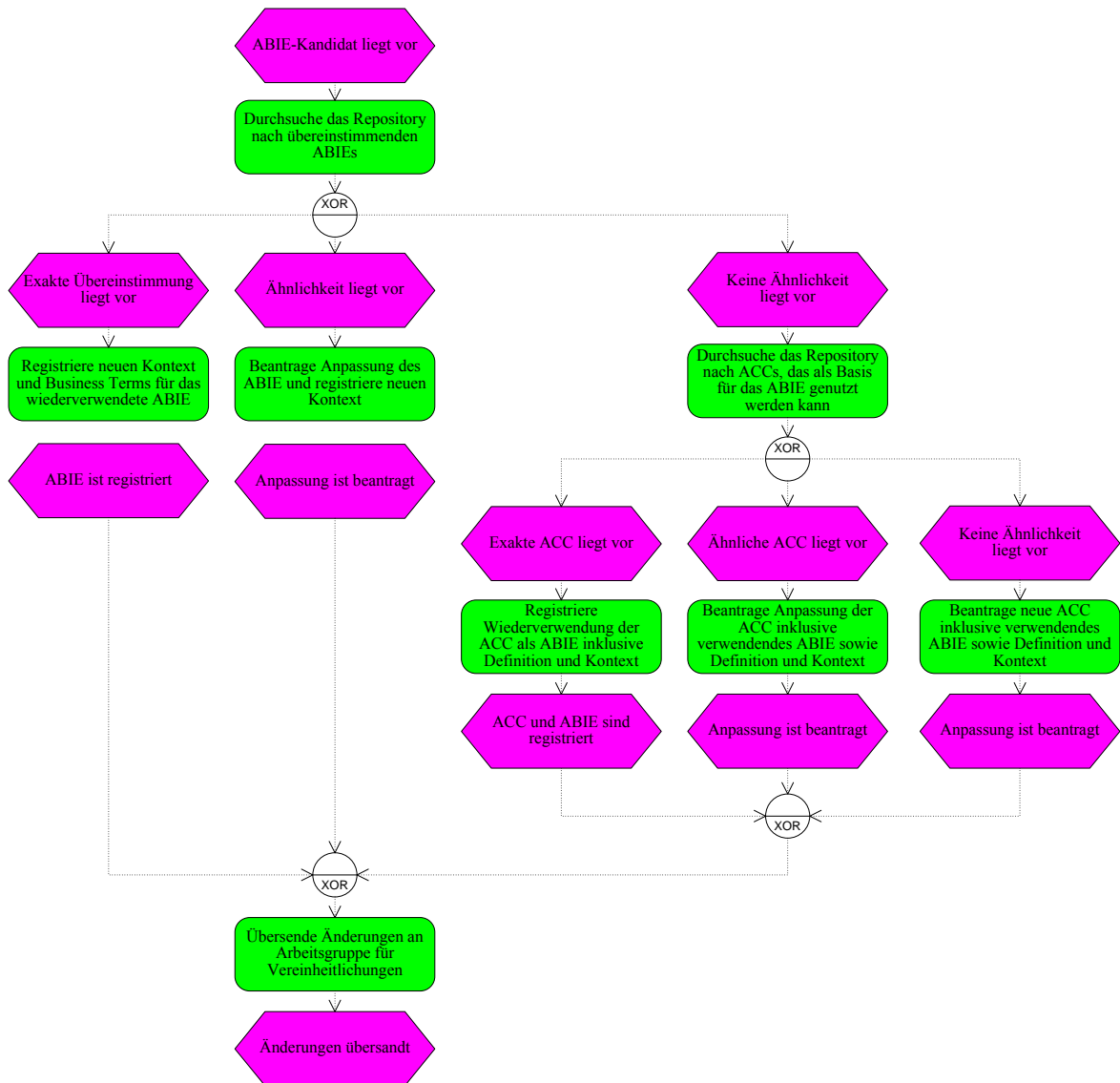


Abb. 52: Vergleich potenzieller ABIE mit existierenden ABIE innerhalb des Repositories in Form einer EPK

C Fachkonzeptionelle Spezifikation konfigurativer Referenzmodellierungsmechanismen

Für die Beschreibung der folgenden Adaptionsmechanismen wird für Konstrukte der Meta- und Meta-Metamodellebene auf *Entity-Relationship-Modelle (ERM)* mit erweiterter Min-Max-Notation zurückgegriffen.

Modelltypselektion

Für diesen grobgranularen Konfigurationsmechanismus wird auf Meta-Metamodellebene eine entsprechende Relation zwischen den ETs *Modelltypvariante* und *Perspektive* eingeführt (vgl. Abb. 53). *Metamodellelemente* repräsentieren Elementtypen, die dem ET *Teilmetamodell* zugeordnet wird. Innerhalb dieses Typs werden abgegrenzte Bereiche definiert, in denen jeweils die Sprache für einen Modelltyp definiert ist, also das Maximum der zulässigen Elementtypen enthält. Den Teilmetamodellen sind zusätzlich *Modelltypvarianten* zugeordnet, um die Modifikation von Eigenschaften zu ermöglichen. Pro Modelltyp existiert grundsätzlich eine Standard-Modelltypvariante, die sämtliche für sie definierte Elementtypen enthält. Modelltypvarianten sind dabei eindeutig einem Modelltyp zugeordnet. Der Konfigurationsmechanismus ordnet die Modelltypvarianten den für sie relevanten *Perspektiven* zu.¹⁸⁴ In Abb. 53 wird bspw. das Fachbegriffsmodell für die Anwendungssystemgestaltung und das ERM für die Organisationsgestaltung ausgeblendet.

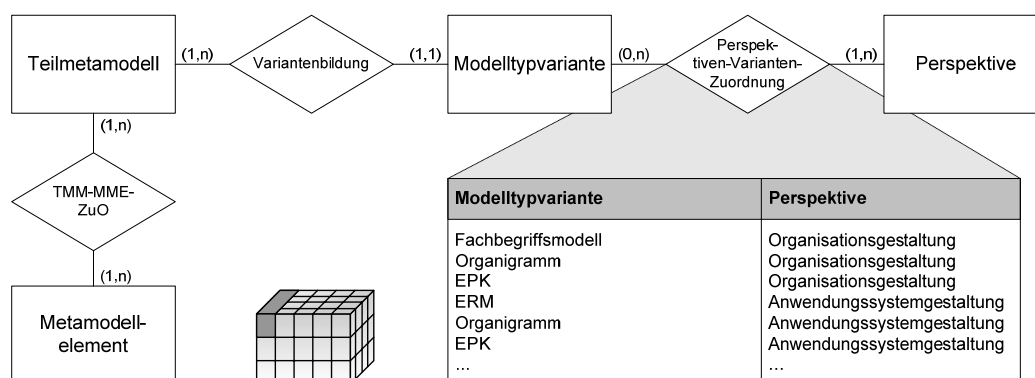


Abb. 53: Meta-Metamodellkonstrukte zur Modelltypselektion¹⁸⁵

¹⁸⁴ Vgl. Becker; Delfmann; Knackstedt (2004b), S. 254.

¹⁸⁵ Quelle: Delfmann (2006), S. 97.

Elementtypselektion

Zur Realisierung der Elementtypselektion wird auf Meta-Metamodellebene eine neue Beziehung zwischen den uminterpretierten Relationshiptypen *TMM-MME-Zuo* und *Modelltypvariantenbildung* eingeführt (vgl. im Folgenden Abb. 54). Innerhalb dieses neuen Relationshiptypen *MTV-MME-Beschränkung* wird die ursprüngliche Zugehörigkeit eines Elementtyps zu einem Modelltyp innerhalb einer Modellvariante wieder aufgehoben. Da eine Modelltypvariante eindeutig zu einem Teilmetamodell gebildet wird, müssen je Ausprägung der MTV-MME-Beschränkung die Einträge des Teilmetamodells übereinstimmen. Die Ausblendung nicht relevanter Elementtypen erfolgt analog zu der Modelltypselektion durch die Zuordnung der Modelltypvarianten zu Perspektiven.¹⁸⁶ Der Einsatz dieses Mechanismus ist bspw. bei erweiterten ereignisgesteuerten Prozessketten sinnvoll, bei denen an Funktionen annotierbare Sprachelemente perspektivenspezifisch variieren können.¹⁸⁷ So ist für die Perspektive Organisationsgestaltung die Annotation von Fachbegriffen sinnvoll, während für die Anwendungssystemgestaltung ETs oder Datenmodelle präferiert werden.

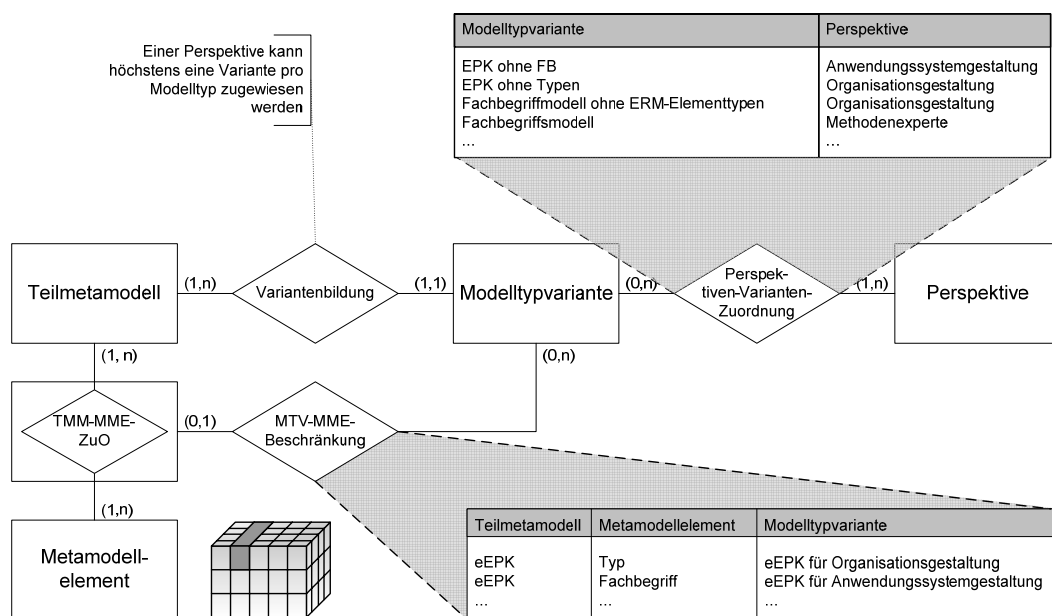


Abb. 54: Meta-Metamodellkonstrukte zur Elementtypselektion¹⁸⁸

Elementsselektion

Die Elementsselektion über Typen, mittels Hierarchiestufenaggregation und über Attribute operieren auf Ausprägungen der Metamodell-elemente. Daher wird auf Meta-Metamodellebene mit der Einführung des ETs *Restriktion WHERE-Klausel* die Möglich-

¹⁸⁶ Vgl. Delfmann (2006), S. 106.

¹⁸⁷ Vgl. Becker; Delfmann; Knackstedt (2004b), S. 254.

¹⁸⁸ In Anlehnung an Delfmann (2006), S. 105.

keit für Instanzrestriktionen für Typen auf Metamodellebene geschaffen (vgl. Abb. 55). Derartige Restriktionen sind abhängig von den jeweiligen Konfigurationsparameterausprägungen, daher sind sie im Meta-Metamodell durch die drei Relationshiptypen *ES über Attribute*, *ES über Typen* und *ES mittels Hierarchies*. mit dem *ET Konfigurationsparameter* verbunden. Die Spezialisierung *Attribut für ES nach Attributen* ermöglicht die explizite Kennzeichnung von Attributen für die Elementselektion auf Metamodellebene. Deren Ausprägungen auf Modellebene werden bei entsprechender Kennzeichnung ebenfalls eliminiert.

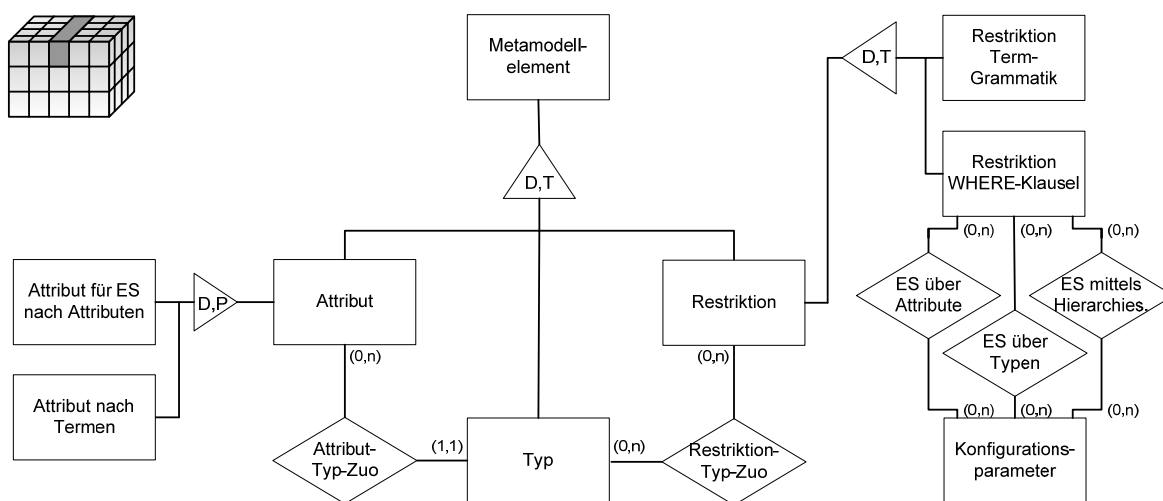


Abb. 55: Meta-Metamodellkonstrukte zur Elementselektion nach Termen¹⁸⁹

Bei der Konfiguration nach Termen hängt die Relevanz des jeweiligen Elements von der Auswertung des zugewiesenen Konfigurationsterms ab. Daher wird auf Meta-Metamodellebene die Spezialisierung *Attribut nach Termen* für Typen auf Metamodellebene eingeführt. Zusätzlich ist die Formulierung einer Restriktion durch den *ET Restriktion Term-Grammatik* erforderlich, um die formale Korrektheit des Terms innerhalb der verwendeten Grammatik sicherzustellen. Auf Metamodellebene werden alle Modellelemente, für die eine termgebundene Adaption vorgesehen ist, zu einem Element aggregiert (vgl. Abb. 56). Diese Generalisierung enthält als Attribut den Konfigurationsterm, dessen Korrektheit durch die formulierten Restriktionen auf Meta-Metamodellebene sichergestellt ist.

¹⁸⁹ Quelle: Becker; Delfmann; Knackstedt (2004b), S. 256.

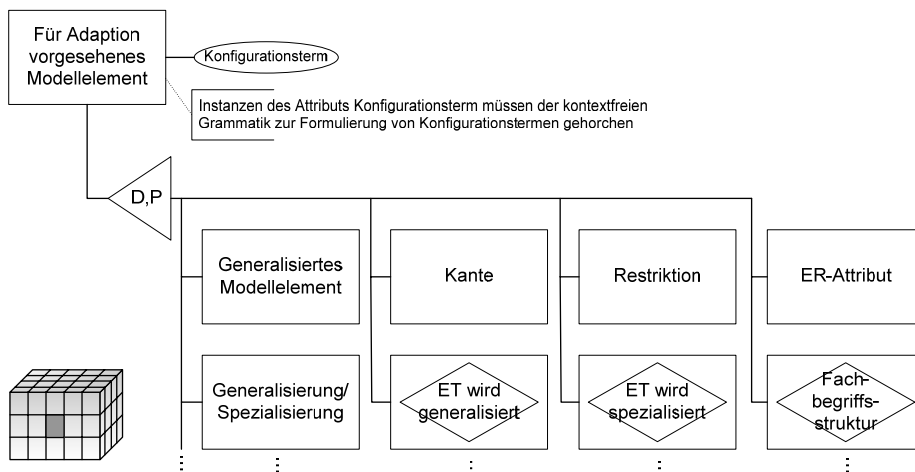


Abb. 56: Metamodellkonstrukte zur Konfiguration nach Termen¹⁹⁰

Bezeichnungsvariation

Für die fachkonzeptionelle Spezifikation der Bezeichnungsvariation wird auf Meta-Metamodellebene der ET *Synonymgruppe* eingeführt, der synonyme Bezeichnungen zusammenfasst (vgl. Abb. 57). Die *Bezeichnungsvariation* weist einer Synonymgruppe entsprechende *Konfigurationsparameter* zu. Die zu verwendende Bezeichnung wird dabei anhand des Attributs *Synonym* erkannt und innerhalb des Modells ausgetauscht. Da die Bezeichnungsvariation nur auf Modellelementen operiert, in denen textuelle Beschreibungen enthalten sind, werden diese Attribute ausgewiesen. Dies geschieht mittels der Spezialisierung des ETs *Attribut* in den ET *Textuelle Beschreibung*.¹⁹¹

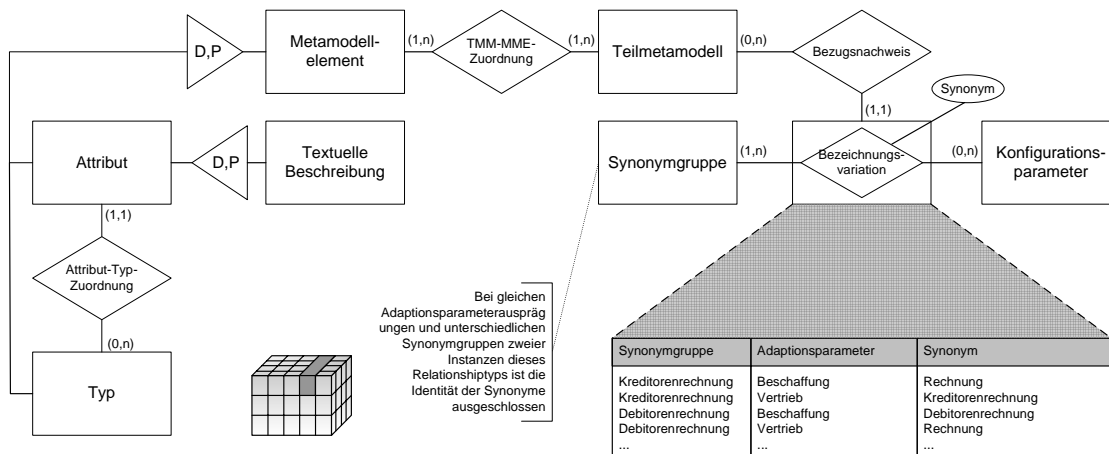


Abb. 57: Meta-Metamodellkonstrukte zur Bezeichnungsvariation¹⁹²

¹⁹⁰ In Anlehnung an Delfmann (2006), S. 126 .

¹⁹¹ Vgl. Delfmann (2006), S. 151.

¹⁹² In Anlehnung an Delfmann (2006), S. 151.

Darstellungsvariation

Die *Darstellungsvariation der Symbole* wird auf Meta-Metamodellebene durch die *perspektivenspezifische Zuordnung von Symbolen zu Metamodellelement-Teilmetamodell-zuordnungen* realisiert (vgl. Abb. 58). Bei der *Darstellungsvariation der Topologie* wird auf Meta-Metamodellebene jeder Kombination von *Modelltypvariante* und *Perspektive* eine spezifische *Topologie* zugewiesen (vgl. Abb. 58). Die Zuordnung der Topologie an Modelltypvarianten anstelle von Teilmetamodellen liegt darin begründet, dass eine Topologie für verschiedene Varianten eines Modelltyps nur begrenzt geeignet ist.¹⁹³ Die *Darstellungsvariation der Konfigurationsansatzpunkte* erfolgt auf Meta-Metamodellebene erfolgt zu der Darstellungsvariation der Topologie mit dem Unterschied, dass der Kombination von *Modelltypvariante* und *Perspektive* statt einer Topologie eine Darstellungsvariante der Konfigurationsansatzpunkte zugewiesen wird (vgl. Abb. 58).¹⁹⁴

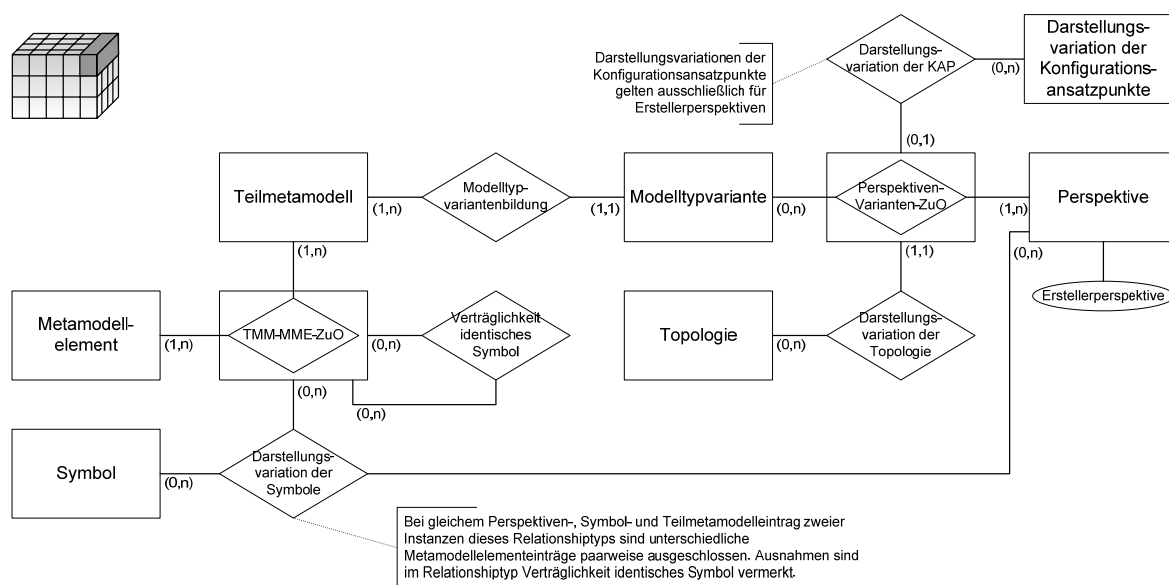


Abb. 58: Meta-Metamodellkonstrukte zur Darstellungsvariation¹⁹⁵

¹⁹³ Vgl. Delfmann (2006), S. 161.

¹⁹⁴ Vgl. Delfmann (2006), S. 161.

¹⁹⁵ Quelle: Delfmann (2006), S. 160.



Institut für Wirtschaftsinformatik
Leonardo-Campus 3
48149 Münster
<http://www.wi.uni-muenster.de>

ISSN 1438-3985