

Entwicklung eines Metamodells für die „Vereinheitlichte Spezifikation von Fachkomponenten“

Peter Fettke, Peter Loos

Johannes Gutenberg-Universität Mainz, Lehrstuhl für Wirtschaftsinformatik und BWL, ISYM - Information Systems & Management, D-55099 Mainz, Germany, Tel.: +49 6131 39-22018, Fax: -22185, E-Mail: {fettke/loos}@isym.bwl.uni-mainz.de, WWW: <http://www.isym.bwl.uni-mainz.de>

Zusammenfassung. Metamodelle sind allgemein bekannte Instrumente zur Systemgestaltung. In diesem Beitrag wird ein Metamodell für die „Vereinheitlichte Spezifikation von Fachkomponenten“ der Arbeitsgruppe 5.10.3 „Komponentenorientierte betriebliche Anwendungssysteme“ der Gesellschaft für Informatik (Memorandum) entwickelt. Das mit Hilfe des Entity-Relationship-Modells formulierte Metamodell enthält weit mehr als 50 Konstrukte (Entitäts- und Beziehungstypen). Im Metamodell können verschiedene Beziehungen zwischen den Spezifikationsebenen des Memorandums aufgezeigt werden. Ebenso ist die Identifikation von Interpretationsspielräumen möglich. In künftigen Arbeiten können das vorgestellte Metamodell verfeinert sowie die unterbreiteten Gestaltungsspielräume zur Weiterentwicklung des Memorandums aufgegriffen werden.

Schlüsselworte: Komponenten, Metaisierung, Komponenten-Repository, Lehre, Modellierung

1 Ausgangssituation und Motivation

Die komponentenbasierte Softwareentwicklung verspricht die Effektivität und Effizienz der Systementwicklung zu verbessern [Same97; Szyp02; Grif98; Turo01]. Während notwendige Technologien für eine komponentenbasierte Entwicklung weitgehend verfügbar sind, bspw. Enterprise Java Beans von Sun oder .NET von Microsoft, besteht ein Entwicklungsbedarf bei der Spezifikation von Fachkomponenten. Einen Beitrag zur Überwindung dieser Problematik leistet das von der Arbeitsgruppe 5.10.3 „Komponentenorientierte betriebliche Anwendungssysteme“ der Gesellschaft für Informatik entwickelte Memorandum „Vereinheitlichte Spezifikation von Fachkomponenten“ [Acke+02] (siehe Abschnitt 2.1, das Wort „Memorandum“ bezeichnet im Folgenden diesen Spezifikationsvorschlag).

Das Memorandum schlägt eine Spezifikation von Fachkomponenten auf unterschiedlichen Spezifikationsebenen vor, wobei unterschiedliche Notationen für die einzelnen Spezifikationsebenen eingeführt werden (bspw. Interface Definition Language (IDL) der Object Management Group (OMG), Object Constraint Language (OCL), Tabellenform). Auf den ersten Blick lässt die gewählte aspektorientierte Spezifikation einer Fachkomponente vermuten, dass die einzelnen Spezifikationsebenen unabhängig voneinander sind. Indes zeigt eine detailliertere Analyse, dass zwischen den einzelnen Konstrukten unterschiedlicher Spezifikationsebenen Abhängigkeiten bestehen können. Beispiele für Abhängigkeiten sind: 1. Auf der Verhaltensebene können nur diejenigen Dienste spezifiziert werden, die auf der Schnittstellenebene bereits definiert worden sind. 2. Die Dienste, die auf der Schnittstellenebene eingeführt worden sind, werden Aufgaben zugeordnet, die auf der Aufgabenebene definiert werden. 3. Gene-

rell sollen auf der Terminologieebene sämtliche Begriffe definiert werden, die auf den anderen Spezifikationsebenen verwendet werden.

Die genannten Beispiele zeigen, dass die vom Memorandum eingeführten Spezifikationsebenen einer Fachkomponente nicht vollständig unabhängig voneinander sind, sondern gegenseitige Abhängigkeiten untereinander aufweisen. Aus Sicht der Autoren leidet hierdurch einerseits die Verständlichkeit für die einzelnen Spezifikationsebenen einer Fachkomponente. Andererseits, kann dies dazu führen, dass die Spezifikation einer Fachkomponente Inkonsistenzen, Redundanzen bzw. Unvollständigkeiten aufweist, die bei separater Betrachtung einzelner Spezifikationsebenen *nicht* ersichtlich werden, sondern erst bei einer ebenenübergreifenden Analyse aufgedeckt werden können.

Um Klarheit über diese Sachverhalte zu schaffen, halten die Autoren es für sinnvoll, ein einheitliches und integriertes Metamodell für die im Memorandum vorgeschlagenen Notationen zu entwickeln (vgl. die Argumentation von [SüEb97, 1f.] im Kontext der Booch-Methode). In einem Metamodell können die einzelnen Konstrukte der verwendeten Notationen und deren Zusammenhang verdeutlicht werden. Die verschiedenen Notationen sollten in einem Metamodell integriert werden, sodass vorhandene Abhängigkeiten und Beziehungen zwischen den Konstrukten der verschiedenen Notationen expliziert werden.

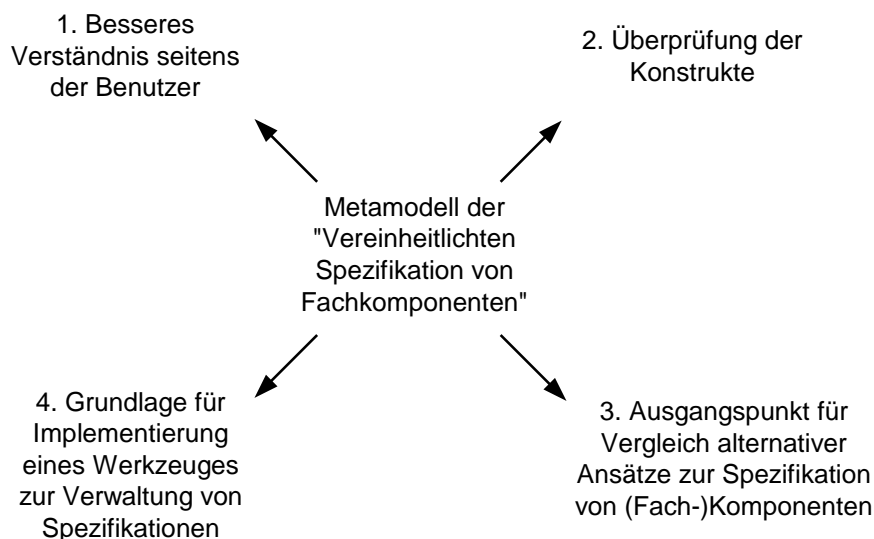


Bild 1: Nutzenpotenziale des Metamodells

Aus einer allgemeinen Perspektive eröffnet ein integriertes Metamodell für die einzelnen Notationen des Memorandums verschiedene Nutzenpotenziale (vgl. Bild 1):

1. Besseres Verständnis für die verschiedenen Konstrukte der Notationen des Memorandums: Ein Metamodell kann die Kommunikation zwischen Nutzern des Memorandums unterstützen, indem sämtliche Konstrukte der Spezifikationsrahmens und ihrer Beziehungen expliziert werden. Hierdurch kann ebenso die Lehrbarkeit des Spezifikationsansatzes in Theorie und Praxis erleichtert werden. Durch Verweis auf bestimmte Konstrukte des Metamodells, kann ferner die Weiterentwicklung des Memorandums und seiner Spezifikationsaspekte systematisch unterstützt werden.
2. Überprüfung der Konstrukte der Notationen des Memorandums: Durch die Erstellung eines Metamodells können Zusammenhänge zwischen verschiedenen Konstrukten der

verschiedenen Notationen aufgedeckt werden. Hierdurch wird es möglich, Konsistenz und Widerspruchsfreiheit zu überprüfen. Ebenso können Abhängigkeiten und Zusammenhänge zwischen Konstrukten verschiedener Ebenen aufgedeckt werden.

3. Ausgangspunkt für Vergleich alternativer Ansätze zur Spezifikation von (Fach-)Komponenten: Neben dem Memorandum existieren alternative Ansätze zur Spezifikation von (Fach-)Komponenten [DSWi98; BJPW99; Fisc00; Han99]. Auf der Grundlage der Metamodelle der verschiedenen Ansätze wird es möglich, Ähnlichkeiten und Gemeinsamkeiten zwischen den Ansätzen systematisch darzustellen und zu untersuchen.
4. Grundlage für die Implementierung eines Werkzeuges zur Verwaltung von Spezifikationen: Ein Metamodell der Notationen des Memorandums kann als ein Ausgangspunkt für die Entwicklung eines Fachkonzeptes eines Werkzeuges zur Verwaltung von Komponentenspezifikationen verwendet werden.

Ziel des Beitrages ist es, ein Metamodell für die im Memorandum definierten Notationen zu entwickeln. Der Beitrag ist folgendermaßen aufgebaut: Nach dieser Einleitung werden im nächsten Abschnitt das Memorandum, der Begriff des Metamodells und verwandte Forschungsarbeiten knapp rekapituliert. Abschnitt 3 stellt das entwickelte Metamodell vor. Im abschließenden vierten Abschnitt werden verschiedene Schlussfolgerungen gezogen und ein Ausblick auf weitere Fragestellungen gegeben.

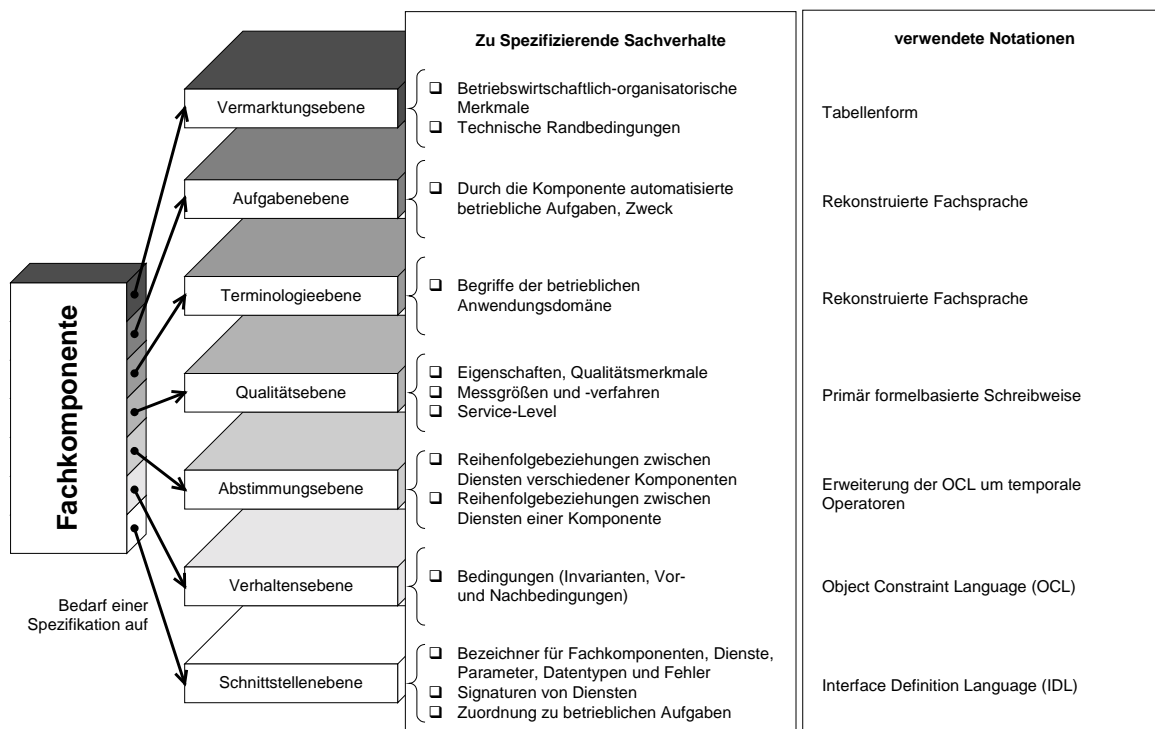


Bild 2: Spezifikationsebenen und –aspekte sowie verwendete Notationen [Acke+02]

2 Theoretischer Hintergrund

2.1 Vereinheitlichte Spezifikation von Fachkomponenten

Das Memorandum verwendet sieben Ebenen zur Spezifikation (Bild 2) [Acke+02]. Jede Ebene fokussiert einen bestimmten Aspekt der Spezifikation einer Fachkomponente und berücksichtigt die Anforderungen einer Rolle im Entwicklungsprozess. Auf den einzelnen Ebenen werden z.T. unterschiedliche Notationen verwendet (OCL, OMG IDL etc.). Für eine ausführliche Diskussion der verschiedenen Spezifikationsaspekte sowie der verwendeten Notationen sei auf die zitierte Literatur verwiesen.

2.2 Metamodelle

Der Begriff Metamodell wird in der Literatur nicht einheitlich verwendet. Allgemein kann ein Metamodell als ein Modell eines Modells verstanden werden. Aus Sicht des Metamodells wird das beschriebene Modell Objektmodell genannt. Bei der Metamodellbildung können zwei Metaisierungsprinzipien unterschieden werden [Stra96, S. 17-28]:

- Ein sprachbasiertes Metamodell repräsentiert die bei der Konstruktion des Objektmodells verwendete Modellierungssprache.
- Ein prozessbasiertes Metamodell repräsentiert die bei der Konstruktion des Objektmodells durchgeführten Modellierungsschritte.

Für den hier verfolgten Zweck sind ausschließlich sprachbasierte Metamodelle von Relevanz, sodass das Wort „Metamodell“ ausschließlich in diesem Sinne im Folgenden verwendet wird. Demnach stellt ein Metamodell die in einem Modell verwendeten Konstrukte und deren Beziehungen dar. Diese Beziehungen definieren, welche Konstrukte in welchen Zusammenhängen erlaubt sind. Ein sprachbasiertes Metamodell ist vergleichbar mit der Grammatikdefinition einer Programmiersprache.

2.3 Verwandte Forschungsarbeiten

Allgemein werden Metamodelle in unterschiedlichen Forschungsbereichen verwendet. Anwendungsbeispiele finden sich bei der Evaluierung von objektorientierten Analysemethoden [Stra96], der Modell Driven Architecture [KIWB03], der Auswahl von Workflowmanagementsystemen [Mühl99] oder der Darstellung und dem Vergleich von Ontologien [DGR02].

Ein spezielles Metamodell für das Memorandum ist den Autoren nicht bekannt, obgleich für einzelne Sichten bereits syntaktische Regelwerke existieren, die als Metamodelle verstanden werden können (Bild 3). Indes sind vorhandene Arbeiten bisher nicht in einem Gesamtmodell integriert dargestellt. In der Arbeit von [Turo01] wird auf allgemeiner Ebene der Zusammenhang zwischen den Begriffen Fachkomponente, Dienst, Schnittstelle, betriebliche Aufgabe, betriebliches Anwendungssystem, Implementierung und Eigenschaft mit Hilfe eines UML-Klassendiagramms geklärt. Dieses Modell kann als ein grobes Metamodell verstanden werden, das als Ausgangspunkt für ein detailliertes Metamodell dienen kann.

Spezifikationsebene	Quellen
Vermarktungsebene	[FeLo01]
Aufgabenebene	[Ortn97]
Terminologieebene	[Ortn97]
Qualitätsebene	-
Abstimmungsebene	[CoTu00]
Verhaltensebene	[OMG01b]
Schnittstellenebene	[OMG01a]
alle Ebenen	[Acke+02]

Bild 3: Ausgangspunkte für die Erstellung eines integrierten Metamodells

3 Entwicklung des Metamodells

In diesem Abschnitt wird das Metamodell für das Memorandum entwickelt (Bild 4). Als Modellierungssprache zur Repräsentation des Metamodells wird das Entity-Relationship-Model (ERM) verwendet. Die Auswahl kann dahingehend begründet werden, dass diese Modellierungssprache relativ weit verbreitet und verständlich ist. Es werden im Folgenden die einzelnen Spezifikationsebenen betrachtet, wobei mit der Schnittstellenebene begonnen wird.

Die Schnittstellenebene besteht aus zwei „Interface“-Spezifikationen: Die eine Spezifikation beschreibt die Dienste, die von der Komponente angeboten werden, die andere Spezifikation (interface extern) die Dienste, die von der Komponente nachgefragt werden. Eine „Interface“-Spezifikation wiederum enthält einfache sowie strukturierte Datentypen, Ausnahmezustände sowie verschiedene Dienste. Die Dienste der Schnittstellenebene können einzelnen Aufgaben der Aufgabenebene zugeordnet werden bzw. Datentypen sind mit Begriffen auf der Terminologieebene zu assoziieren.

Auf der Verhaltensebene werden Bedingungen definiert, die das Verhalten einzelner Dienste festlegen. Auf der Ebene können eine Reihe von Bedingungen spezifiziert werden, wobei jede einzelne Bedingung eindeutig einer Komponente zuzuordnen ist. Der Kontext einer Bedingung kann durch die Angabe eines Dienstes präzisiert werden, auf den sich die Bedingung bezieht. Zur Formulierung der Bedingungen können die auf der Schnittstellenebene verwendeten Datentypen verwendet werden. Die angeführten Beziehungen zu Diensten und Datentypen der Schnittstellenebene dürfen sich nur auf die eigene und nicht auf die externe Schnittstelle beziehen. Diese Bedingung ist nicht explizit im Metamodell formuliert.

Die Abstimmungsebene besteht aus einer Reihe von Bedingungen, die sich zwingend auf einen Dienst der Schnittstellenebene beziehen. Neben der Kontextdefinition einer Bedingung, kann eine Bedingung weiterhin Notwendigkeiten zur Abstimmung zu anderen Diensten der Komponente definieren (Intra-Abstimmungsbedingung). Andererseits können Abstimmungen zu Diensten anderer Komponenten beschrieben werden (Inter-Abstimmungsbedingung). Während die Beziehungen „Kontext“ und „Intra-Abstimmung“ zwingend auf Dienste der eigenen Schnittstelle verweisen, bezieht sich die Beziehung „Inter-Abstimmung“ zwingend auf Dienste der externen Schnittstelle (diese Bedingung ist nicht im Metamodell formuliert).

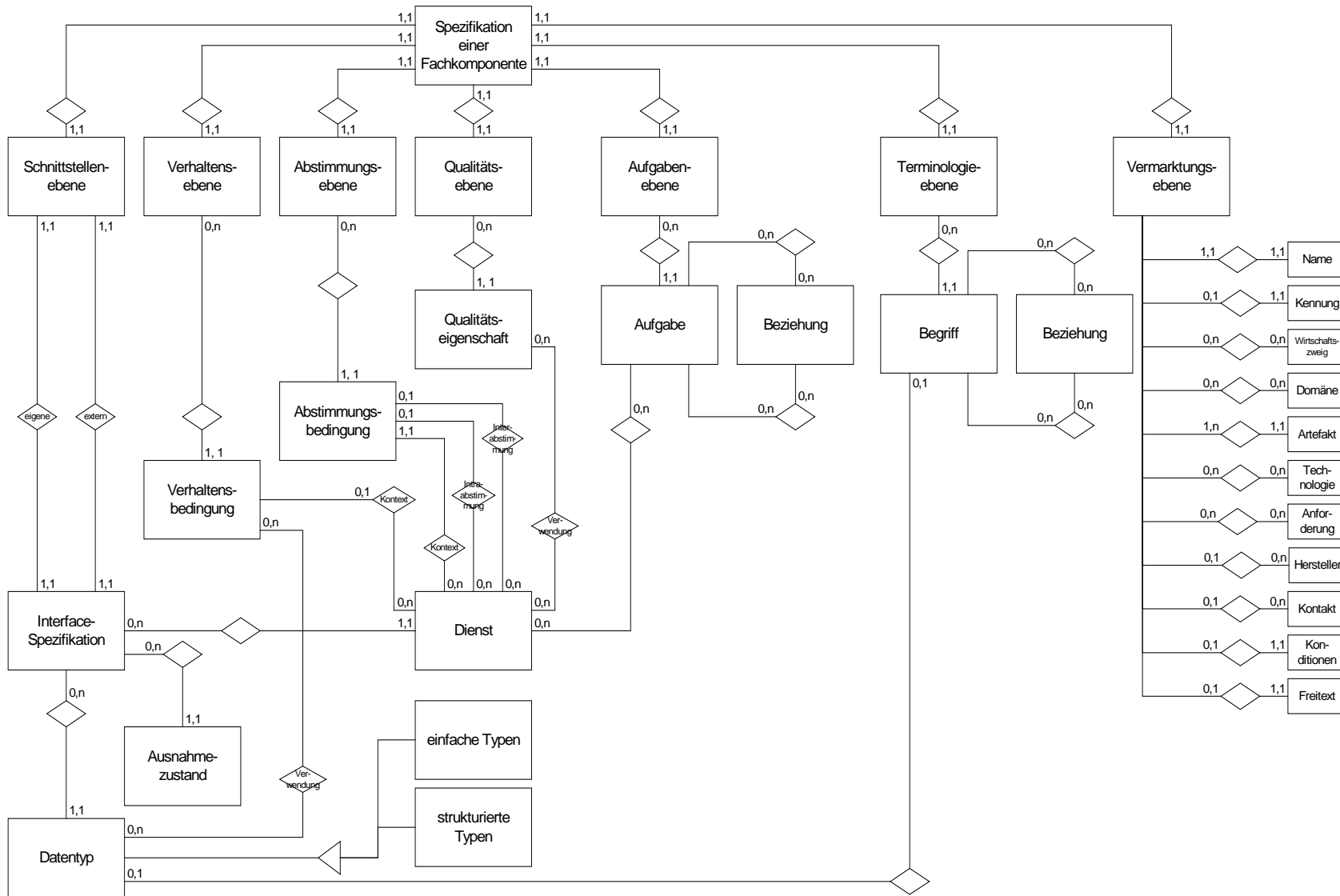


Bild 4: Metamodell für die Notationen des Memorandums

Die zur Zeit im Memorandum beschriebene Verfahrensweise zur Spezifikation der Qualitätsebene enthält nur wenige Hinweise auf erlaubte Spezifikationskonstrukte. Wird einer relativ allgemeinen Auffassung gefolgt, kann die Qualitätsebene als eine Menge von Qualitätseigenschaften verstanden werden. Einzelne Qualitätseigenschaften können hierbei speziellen Diensten einer Komponente zugeordnet werden, die sowohl durch die eigene als auch die externe Schnittstelle definiert werden. Auf diese Weise ist es auch möglich zu spezifizieren, dass die Dienste, die von einer Komponente nachgefragt werden, bestimmten Qualitätseigenschaften zu genügen haben.

Die Metamodellteilbereiche für die Aufgaben- und Terminologieebene sind analog aufgebaut. Es wird jeweils auf eine explizite Darstellung von Satzbauplänen etc. verzichtet, um das entwickelte Metamodell überschaubar zu halten. Während auf der Aufgabenebene eine Reihe von Aufgaben zuzüglich Beziehungen zwischen den Aufgaben definiert werden, sind auf der Terminologieebene Begriffe und Begriffsbeziehungen zu beschreiben. Mögliche Beziehungen zwischen Aufgaben sind bspw. „ist alternative Aufgabe“ oder „ist vorgelagert“. Beziehungen zwischen Begriffen können bspw. sein „ist synonym“, „ist teil-synonym“ und „ist Beispiel für“.

Die einzelnen Spezifikationsattribute der Vermarktungsebene können intuitiv sowohl als ERM-Attribute einer Komponente (bspw. Name, Kennung) als auch als eigenständige Entitätstypen (bspw. Hersteller, Ansprechpartner) verstanden werden. Um eine einheitliche Übersetzung des Memorandums in das Metamodell vorzunehmen und eine – aus Sicht der Autoren – wenig zielführende Diskussion zu vermeiden, ob bestimmte Informationen als „Entitäten“ oder als „Attribute“ von Entitäten zu verstehen sind, wurden sämtliche Spezifikationsattribute dieser Ebene als Entitätstypen im Metamodell repräsentiert. Durch die Festlegung unterschiedlicher Kardinalitäten wird ausgedrückt, ob es sich um optionale oder verpflichtende Angaben handelt bzw. ob einzelne Informationen für verschiedene Komponenten mehrfach verwendet werden können.

4 Schlussfolgerungen und Ausblick

In diesem Beitrag wurde ein relativ grobes Metamodell des Memorandums vorgestellt, das trotzdem mehr als 50 Konstrukte (Entitäts- und Beziehungstypen) enthält. Aus dem Metamodell wird ersichtlich, dass einzelne Konstrukte einer Komponentenspezifikation eine zentrale Bedeutung besitzen. Beispielsweise werden Dienste der Schnittstellenebene auf der Verhaltens-, Abstimmungs-, Aufgaben- und evtl. der Qualitätsebene – zum Teil sogar mehrfach – referenziert. Anderen Konstrukten kommt dagegen nur eine geringere Bedeutung zu. Im Memorandum wird bspw. ausgeführt, dass auf der Terminologieebene die Bedeutung von Begriffen zu klären ist, die auf sämtlichen Ebenen verwendet werden. Allerdings ist aus dem Metamodell ersichtlich, dass nur explizite Beziehungen zwischen Datentypen und Begriffen bestehen. Folglich gibt das Memorandum nur wenige Gestaltungsrestriktionen bzw. -hinweise, welche Begriffe auf der Terminologieebene zwingend zu erläutern sind.

Im Rahmen der Untersuchung sind verschiedene Grenzen deutlich geworden, die im Folgenden angesprochen werden:

- Subjektivität der Metamodellbildung: Das Bilden von Metamodellkonstrukten ist nicht immer eindeutig. Bestimmte Sachverhalte können aus unterschiedlichen Perspektiven unterschiedlich dargestellt werden. Beispielsweise ist aus dem Memorandum nicht ersichtlich, ob zu einer Aufgabe auf der Aufgabenebene zwingend ein

Dienst auf der Spezifikationsebene angegeben werden muss oder ob dieser optional ist. Dieser Sachverhalte sollte in einer künftigen Memorandumsversion geklärt werden.

- Detaillierung der Abbildung: Es ist nicht immer offensichtlich, wie detailliert das Metamodell zu erstellen ist. Beispielsweise wurde im vorgestellten Metamodell ein Dienst auf der Schnittstellenebene durch jeweils ein Entitätstyp repräsentiert. Ebenso wäre es denkbar, das Metamodell weiter zu detaillieren, indem darüber hinaus die Signaturen eines Dienstes im Metamodell repräsentiert werden.
- Verwendete Metamodellierungssprachen: Als Metamodellierungssprache wird das ERM verwendet. Es stellt sich die Frage, ob die Verwendung anderer Metamodellierungssprachen zu Vorteilen führt, da diese evtl. eine höhere Ausdrucksmächtigkeit besitzen. Beispielsweise konnten in der verwendeten Modellierungssprache einzelne Randbedingungen nicht im Metamodell, sondern nur natürlichsprachlich formuliert.
- Im vorgestellten Metamodell wurde ausschließlich die primäre Notation berücksichtigt, ohne auf Eigenarten der sekundären Notation einzugehen. Es ist zu überlegen, inwieweit die sekundäre Notation Erweiterungen und Ergänzungen am vorgestellten Metamodell notwendig werden lässt. Beispielsweise wäre es denkbar, für sämtliche Wörter einer natürlichsprachlichen Spezifikation – wie sie bspw. für die Abstimmungsebene vorgeschlagen wird – zu fordern, dass sie ebenso auf der Terminologieebene definiert werden. Andererseits kann der Standpunkt vertreten werden, dass bei Verwendung der sekundären Notation generell nicht auf eine hohe Konsistenz zwischen den Abstimmungsebenen zu achten ist und daher sekundäre Notationen prinzipiell nicht im Metamodell zu berücksichtigen sind.

Die Autoren schlagen einerseits vor, die bei der Untersuchung aufgedeckten Probleme bei der Überarbeitung und Weiterentwicklung des Memorandums zu berücksichtigen. Andererseits kann das hier eingeführte Metamodell weiter gepflegt und Bestandteil einer künftigen Memorandumsversion werden, sodass die einleitend angeführten Nutzenpotenziale weitgehend realisiert werden können.

Literatur

[Acke+02] *Ackermann, J.; Brinkop, F.; Conrad, S.; Fettke, P.; Frick, A.; Glistau, E.; Jaekel, H.; Kotlar, O.; Loos, P.; Mrech, H.; Raape, U.; Ortner, E.; Overhage, S.; Sahm, S.; Schmietendorf, A.; Teschke, T.; Turowski, K.* Vereinheitlichte Spezifikation von Fachkomponenten - Memorandum des Arbeitskreises 5.10.3 Komponentenorientierte betriebliche Anwendungssysteme. <http://wi2.wiwi.uni-augsburg.de/gi-memorandum.php.htm>, Abruf 2003-03-01.

[BJPW99] *Beugnard, A.; Jézéquel, J.-M.; Plouzeau, N.; Watkins, D.*: Making Components Contract Aware. In: *IEEE Computer* 32 (1999) 7, S. 38-45.

[CoTu00] *Conrad, S.; Turowski, K.*: Vereinheitlichung der Spezifikation von Fachkomponenten auf der Basis eines Notationsstandards. In: *J. Ebert; U. Frank (Hrsg.): Modelle und Modellierungssprachen in Informatik und Wirtschaftsinformatik - Beiträge des Workshops "Modellierung 2000"*, St. Goar, 5.-7. April 2000. Koblenz 2000, S. 179-194.

[DSWi98] *D'Souza, D. F.; Wills, A. C.*: Objects, Components, and Frameworks with UML - The Catalysis Approach. Reading, MA, et al. 1998.

- [DGR02] *Davies, I.; Green, P.; Rosemann, M.:* Facilitating an Ontological Foundation of Information Systems with Meta Models. In: *A. Wenn; M. McGrath; F. Burstein (Hrsg.):* Proceedings of the 13th Australasian Conference on Information Systems (ACIS 2002), 3-6 December. Melbourne 2002, S. 937-947.
- [FeLo01] *Fettke, P.; Loos, P.:* Ein Vorschlag zur Spezifikation von Fachkomponenten auf der Administrations-Ebene. In: *K. Turowski (Hrsg.):* Modellierung und Spezifikation von Fachkomponenten: 2. Workshop im Rahmen der vertIS (verteilte Informationssysteme auf der Grundlage von Objekten, Komponenten und Agenten) 2001, Bamberg, Deutschland, 05. Oktober 2001. Bamberg 2001, S. 95-104.
- [Fisc00] *Fischer, B.:* Specification-Based Browsing of Software Component Libraries. In: *Journal of Automated Software Engineering* 7 (2000) 2, S. 179-200.
- [Grif98] *Griffel, F.:* Componentware - Konzepte und Techniken eines Softwareparadigmas. Heidelberg 1998.
- [Han99] *Han, J.:* An Approach to Software Component Specification. Proceedings of 1999 International Workshop on Component Based Software Engineering. Los Angeles, USA 1999.
- [KIWB03] *Kleppe, A.; Warmer, J.; Bast, W.:* MDA Explained: The Model Driven Architecture: Practice and Promise. Boston et al. 2003.
- [Mühl99] *Mühlen zur, M.:* Evaluation of Workflow Management Systems Using Meta Models. Proceedings of the 32th Hawaii International Conference on Systems Science (HICSS '99). Hawaii 1999.
- [OMG01a] *OMG:* The Common Object Request Broker: Architecture and Specification: Version 2.5. Framingham 2001.
- [OMG01b] *OMG:* Unified Modeling Language Specification: Version 1.4. Needham 2001.
- [Ortn97] *Ortner, E.:* Methodenneutraler Fachentwurf - Zu den Grundlagen einer anwendungsorientierten Informatik. Stuttgart, Leipzig 1997.
- [Same97] *Sametinger, J.:* Software Engineering with Reusable Components. Berlin et al. 1997.
- [Stra96] *Strahringer, S.:* Metamodellierung als Instrument des Methodenvergleichs - Eine Evaluierung am Beispiel objektorientierter Analysemethoden. Aachen 1996.
- [SüEb97] *Süttenbach, R.; Ebert, J.:* A Booch Metamodel. Fachberichte Informatik 5/97. Koblenz 1997.
- [Szyp02] *Szyperski, C.:* Component Software - Beyond Object-Oriented Programming. 2. Aufl., London et al. 2002.
- [Turo01] *Turowski, K.:* Fachkomponenten - Komponentenbasierte betriebliche Anwendungssysteme. Habil.-Schr. Magdeburg 2001.