

# **Eine Grundlegung des Rollenbegriffs für die objektorientierte Modellierung mit dem Vorschlag einer Änderung von UML\***

Friedrich Steimann

## **Zusammenfassung**

Rollen gewinnen in der objektorientierten Modellierung zunehmend an Bedeutung. Dabei waren Rollen eigentlich schon immer da – eine einheitliche, klare Definition des Rollenbegriffs und deren Umsetzung in gängige Modellierungsformalismen fehlt jedoch bis heute. Beginnend mit ontologischen und linguistischen Betrachtungen sowie mit einer kurzen Darstellung des Rollenbegriffs in der Datenmodellierung wird eine solche Definition formal herausgearbeitet und auf UML, die derzeit am meisten diskutierte der objektorientierten Modellierungssprachen, übertragen.

## **1 Einleitung**

Um es gleich vorwegzunehmen: Die Rolle ist für die objektorientierte Modellierung ein so fundamentales Konzept, daß sie als auf einer Ebene mit anderen wie Klasse (oder Typ) und Relation stehend angesehen werden muß. Die Tatsache, daß sie dennoch so wenig Verbreitung gefunden hat, liegt vor allem darin begründet, daß in der Praxis gar nicht zwischen Rolle und Klasse unterschieden wird, sondern Klassen für beide Arten von Konzepten verwendet werden. Tatsächlich ist die Unterscheidung zwischen Rolle und Klasse vor allem eine inhaltliche; man kann syntaktisch recht gut ohne sie leben. Semantisch bereitet die Verschmelzung der beiden Konzepte allerdings erhebliche Probleme.

Die Arbeit beginnt mit einer Bestimmung des Rollenbegriffs aus den Perspektiven dreier Disziplinen: der der Ontologie, der der Linguistik und der der Datenmodellierung. Daran anschließend wird eine einfache formale Modellierungssprache vorgestellt, die einen Rollenbegriff beinhaltet, der vor allem den ontologischen Anforderungen genügt. Dieser Begriff wird dann auf die objektorientierte Softwaremodellierung übertragen, die ebenfalls Rollen kennt, aber derzeit noch nicht mit einer befriedigenden Definition aufwarten kann. Abschließend wird eine entsprechende Änderung der UNIFIED MODELING LANGUAGE (UML) vorgeschlagen.

## **2 Begriffsbestimmung**

### **2.1 Rolle als ontologisches Konzept**

Was also ist eine Rolle? Die wohl grundlegendste Antwort habe ich bei Guarino [Guarino 1992] gefunden. Er definiert Rolle (in Abgrenzung vom natürlichen Typ) als ein Konzept, das fundiert, aber nicht semantisch rigide ist. Die erste Eigenschaft besagt, daß keine der Instanzen eines solchen Konzeptes für sich allein existieren kann, daß also jede Instanz, um unter das Konzept zu fallen, in Beziehung zu mindestens einer anderen stehen muß. Dabei sind Teil-Ganzes-Beziehungen ausdrücklich ausgeschlossen. So ist *Student*

\* erschienen in: J Ebert, U Frank (Hrsg) *Modelle und Modellierungssprachen in Informatik und Wirtschaftsinformatik: Modellierung 2000* (Fölbach-Verlag, Koblenz 2000) 55–69.

z.B. fundiert, weil ein Student, der keinem Studium nachgeht, kein Student ist. Ein Rad dagegen ist nicht fundiert, obwohl es in der Regel Teil von etwas anderem ist und selbst aus Teilen besteht.

Die zweite Eigenschaft, die mangelnde semantische Rigidität, verlangt, daß eine Rolle ein Konzept sein muß, das nichts zur Identität seiner Instanzen beiträgt, eine Instanz also gewissermaßen der Extension dieses Konzeptes beitreten und diese wieder verlassen kann, ohne daß sich das auf die Identität der Instanz auswirkt. Das Konzept *Kleinkind* beispielsweise ist nicht semantisch rigide, da ein Säugling im Alter von einem Jahr zum Kleinkind (und damit zu einer Instanz des Konzepts *Kleinkind*) wird, dies aber genauso wie die spätere Aufgabe des Konzeptes nichts zur Identität der Person beiträgt. *Person* dagegen ist semantisch rigide, denn wer einmal eine Person ist, wird immer eine Person bleiben (oder muß seine Identität aufgeben). *Student* wiederum ist nicht semantisch rigide; *Student* ist damit eine Rolle. [Guar 1992]

Der größte Nachteil Guarinos Definition ist, daß sie eine exakte Definition der Teil-Ganzes-Beziehung voraussetzt. Diese bleibt jedoch weiterhin schwierig. Außerdem ist die Absolutheit der Definition nur eine scheinbare: Was fundiert ist und was nicht, hängt wesentlich vom Modellierungskontext ab. Dasselbe gilt natürlich auch für die semantische Rigidität. Was bleibt, ist jedoch ein wertvolles Entscheidungskriterium für eine der, wie ich meine, wesentlichsten Entwurfsentscheidungen in der Modellierung.

Auf der anderen Seite ist Guarinos Rollendefinition auch gar nicht praktischer, sondern ontologischer Natur; die Fassung der Fundierung z. B. geht auf die Logischen Untersuchungen Husserls (1859–1938) [Huss 1901] zurück, der als Begründer der formalen Ontologie gilt. Die Definition zielt damit, wie bei ontologischen üblich, auf die Feststellung der Grundbausteine einer Weltsicht ab. Andere Definitionen sind weniger grundsätzlich: Sie versuchen, Rollen in bestehende Modellierungskonstrukte abzubilden oder eine Modellierungssprache um ein definiertes Rollenkonzept erweitern, um Rollen bei der Modellierung nach Gutdünken verwenden zu können. Interessanterweise ist Guarinos Rollendefinition jedoch inhaltlich nicht weit entfernt von denen, die pragmatisch eingeführt werden: Die Bedingung der mangelnden semantischen Rigidität findet man häufig als die Betonung des temporären oder dynamischen Charakters der Rolle wieder, und manchmal die Fundierung als die Tatsache, daß Rollen im Kontext von oder über Relationen definiert werden [Stein 1999].

## 2.2 Rollen in der Linguistik

Der ontologischen Bestimmung des Rollenkonzepts durch Guarino geht dessen Entdeckung in der Sprache um einiges voraus. Eco nennt Lodwick (1619–1694) als den ersten Vorläufer der lexikalischen Semantik:

*“Lodwick hat also die originelle Idee, nicht von den Substantiven auszugehen (den Nomina oder Namen von Individuen und Gattungen, wie es in der aristotelischen Tradition noch bis zu seiner Zeit üblich war), sondern von Handlungsmustern, und er bevölkert dann diese Handlungsmuster mit Akteuren bzw. mit jenen Figuren, die wir heute Aktanten nennen würden, d. h. mit abstrakten Rollen, die sich dann mit Namen von Personen oder Dingen oder Orten verknüpfen lassen als Handelnde, Gegen-Handelnde, Objekte der Handlung und so weiter.”* [Eco 1994, S. 268]

Tatsächlich stellt Lodwick die Verben (die ja für gewöhnlich das Prädikat eines Satzes stellen und damit dessen zentrale Relation benennen) an den Anfang und leitet in seiner

Entwicklung einer Universalsprache “A common writing” die Wörter der anderen Wortformen systematisch davon ab. Seine Systematik führt ihn dann u. a. dazu, Guarinos ontologisch basierte Unterscheidung zwischen natürlichen Typen und Rollen vorwegzunehmen:

*“Next the verbes, follow in order the nounes substantives, of which there are two sorts.*

*Appellative.  
proper.*

*Appellative I thus distinguish. To be a name by which a thing is named and distinguished, but not continually, only for the present, in relation to some action done or suffered, as for instance, Speech being of a murther committed; he that committed the same, will, from the act, be called a murtherer, and the party on whom the act is committed, the murthered, these names thus given in reference to the action done, continues no longer with the party, then thought is had of the action done, but on the contrary the specificall proper name, remaineth continually with the denominated, as the specificall name of man, beast, so also the individuall denomination of any particular man, as Peter, Thomas, &c.*

*A proper name is that, by which any thing is constantly denominated, specifically, as Man, dog, horse.”* [Lodw 1647, S. 7–8]

Man beachte, daß Lodwicks Unterscheidung von “Appellative” und “proper” sowohl Fundierung als auch semantische Rigidität anspricht. “Murtherer” und “murthered” sind fundiert (“these names thus given in reference to the action done” – ungefähr “diese Namen werden in Bezug auf die Handlung vergeben”) und nicht semantisch rigide (“these names ... continues no longer with the party, then thought is had of the action done” – ungefähr “diese Namen verbleiben nicht länger bei den Beteiligten, als man an die Handlung denkt”); sie sind Rollen des Prädikats *morden* (“murther committed”). “*Man, dog, horse*” bezeichnen dagegen natürliche Typen.

Lodwick war Kaufmann und von der Idee beseelt, eine Universalsprache zu erschaffen. Die Suche nach einer Universalsprache, die nicht nur von jedermann gesprochen und verstanden werden kann, sondern die auch ein eindeutiges Abbild der Realität zu formulieren erlaubt, hat später unter anderem zur Entwicklung der symbolischen Logik geführt, die Basis vieler heutiger Modellierungssprachen ist – das Konzept der Rolle taucht darin jedoch nicht mehr auf.

Ganz anders in der Linguistik. Der Sprachtheoretiker Bühler (1879–1963) konstatierte:

*“Es bestehen in jeder Sprache Wahlverwandtschaften; das Adverb sucht sein Verbum und ähnlich die anderen. Das läßt sich auch so ausdrücken, daß die Wörter einer bestimmten Wortklasse eine oder mehrere Lehrstellen um sich eröffnen, die durch Wörter bestimmter anderer Wortklassen ausgefüllt werden müssen.”* [Bühl 1934, S. 173]

Etwa zur gleichen Zeit, doch erst viel später veröffentlicht, entstand Tesnières Dependenzgrammatik [Tesn 1965], nach der ein Wort, das Regens, im Satz andere Wörter, seine Dependenzien, fordert. Die Leerstellen Böhlers sind die Rollen eines Regens, die von dessen Dependenzien gefüllt werden. In Fillmores Case grammar wird dem syntaktischen Kasus, der sog. Kasusform, die sog. Kasusrolle oder auch semantische Rolle gegenüber-

gestellt, die auf den Inhalt der logischen Ergänzung eines Prädikats abzielt. Anstatt es aber bei der Feststellung zu belassen, daß jedes Verb eine (seine eigene) Menge von semantischen Rollen hat, die durch andere Satzglieder (explizite oder implizite) zu füllen sind, besteht er darauf, daß eine grammatische Theorie ein Inventar von abstrakten Rollen bereitstellen muß, auf die sich alle konkreten reduzieren lassen: eben Handelnder, Gegen-Handelnder etc. [Fill 1971]. Aktuelle Bestrebungen, die Grammatik ganz in das Lexikon zu verlagern, gehen wieder davon ab; und auch für die Modellierung der Realität haben diese abstrakten Rollen wenig Wert.

## 2.3 Rollen in der Datenmodellierung

Die klassische Datenmodellierung baut darauf, daß sich die Welt durch große Mengen gleichartiger Daten darstellen läßt. Jede solche Menge wird durch einen Typ oder eine Klasse charakterisiert, und die Instanzen dieses Typs stehen für die Individuen, die die Realität ausmachen.

Bachman und Daya [BaDa 1977] waren vielleicht nicht die ersten, denen auffiel, daß es sich bei den meisten Typen in Datenmodellen eigentlich gar nicht um Typen, sondern um Rollen handelt, aber sie waren wohl die ersten, die sich dem Problem stellten: *Student, Dozent, Angestellter, Kunde, Aktionär* – all das sind verschiedene Rollen, die die Individuen eines einzigen Typs, nämlich *Person*, spielen können. Mehr noch: Manche Personen können mehrere Rollen gleichzeitig spielen (aber jede Instanz muß Instanz genau eines Typs sein), Rollen werden von Individuen angenommen und abgelegt (nicht aber deren Typen) und es gibt Rollen, die von Instanzen mehrerer Typen gespielt werden können. Eine der Grundannahmen der Datenmodellierung, nämlich daß *ein* Objekt der Realität durch *ein* Objekt im Modell repräsentiert wird, wird mit dem Fehlen der Unterscheidung zwischen Rollen und natürlichen Typen negiert. Dem halfen Bachman und Daya durch die Einführung eines geeigneten Rollenbegriffs in das Netzwerkmodell ab.

Seit dieser Zeit hat es zahlreiche Veröffentlichungen zum Thema Rollen in der Daten-, der konzeptuellen und der objektorientierten Modellierung und Programmierung gegeben. Eine umfassende Übersicht zu dem Thema mit einer Besprechung der wichtigsten Ansätze findet sich in [Stei 1999]. Dabei läßt sich der Großteil der Arbeiten grob in drei Bereiche einteilen. Klar abtrennbar ist das *Object Role Model* (ORM), das von Falkenberg [Falk 1976] vorgeschlagen und von Nijssen und Halpin [NiHa 1989] (unter dem Namen NIAM) populär gemacht und zum ER-Konkurrenten ausgebaut wurde. Im ORM wird die Rolle als benannte Stelle einer Relation (oder Prädikation) in den Mittelpunkt der Modellierung gestellt. Bei genauer Betrachtung sind die Unterschiede zum ER-Modell jedoch eher kosmetischer Natur.

Der Rest der Arbeiten ist zweigeteilt: Der eine faßt Rollen als Spezialisierung und/oder Generalisierung auf, verlangt aber in jedem Fall, daß Instanzen dynamisch ihren Typ wechseln können (dynamische Klassifikation oder Instanzmigration), und der andere modelliert eine Rolle über eine beigeordnete Instanz, also durch eine zusätzliche Instanz, die der rollenspielenden gewissermaßen zur Seite gestellt wird. In letzterem Fall wird ein (logisches) Individuum durch mehrere Instanzen (eine für die Identität und eine je Rolle) vertreten. Beide Ansätze sind jedoch nicht frei von konzeptuellen und technischen Problemen; diese werden mit dem im nächsten Abschnitt vorgestellten alternativen Fassung des Rollenbegriffs vermieden [Stei 1999].

Es bleibt zu bemerken, daß, gemessen an Guarinos Rollendefinition, viele Arbeiten wohl das Kriterium der mangelnden Rigidität (die temporäre oder dynamische Natur der

Rollen), jedoch nicht das der Fundierung (die Notwendigkeit einer Relation) zur Begriffsbestimmung heranziehen. Bei diesen Autoren ist also *Kleinkind* durchaus eine Rolle, obwohl es nicht fundiert ist.<sup>1</sup> Eine solchermaßen eingeschränkte Rollendefinition läßt sich vielleicht dadurch erklären, daß manche Autoren den Ursprung des Rollenbegriffs im Theater suchen [Booc 1994]: Die Rolle des Hamlets beispielsweise kann ein Schauspieler annehmen und wieder ablegen, ohne daß es dazu irgendeiner Beziehung zu anderen bedarf. Jedenfalls taugt Guarinos ontologische Untersuchung durchaus auch dazu, diese Art von Konzepten als dritte abzugrenzen: nicht fundiert und nicht semantisch rigide. Man könnte sie vielleicht *dynamische Klasse* oder *Zustand* nennen; es handelt sich jedenfalls nicht um Rollen in dem hier favorisierten Sinn (man vgl. dazu die Arbeit von Wieringa et al. [WdS 1995]).

### 3 Eine formale Rollendefinition

Da die Formalisierung des Rollenbegriffs nicht im leeren Raum erfolgen kann, möchte ich zunächst eine einfache, aber durchaus zweckmäßige Modellierungssprache skizzieren. Ein Modell bestehe demnach aus einer Menge von Typsymbolen  $S$  (für Spezies) wie z. B. *Person*, *Auto* etc. Zu jedem Typsymbol  $s$  aus  $S$  gehöre eine Menge von Individuenbezeichnern  $I_s$ .<sup>2</sup> Mit jedem Typsymbol  $s \in S$  außerdem assoziiert sei

- eine Intension  $\text{int}(s)$ , die die Eigenschaften angibt, die den Individuen des Typs beigegeben werden (die den Typ charakterisieren),
- eine Extension  $\text{ext}(s)$ , die die Individuen des Typs aufzählt, und
- eine Interpretation (oder Denotation), die die Semantik des Typsymbols ausmacht.<sup>3</sup>

Während die Intension eines Typs üblicherweise invariant ist (sie macht ja den Typ aus), ist die Veränderlichkeit der Extension charakteristisch für jedes dynamische zu modellierende System. Man unterscheidet daher zwischen

- der statischen Extension  $\text{ext}(s)$ , die alle jemals existierenden Individuen des Typ  $s$  umfaßt, und
- der dynamischen Extension  $\text{ext}_t(s)$ , die alle zum Zeitpunkt  $t$  existierenden Individuen des Typs umfaßt.

Die statische Extension  $\text{ext}(s)$ , die Teil des statischen Modells ist, entspricht  $I_s$ . Die dynamischen Extensionen  $\text{ext}_t(s)$ , die stets eine Teilmenge von  $\text{ext}(s)$  sind, sind Teil des dynamischen Modells. Erzeugung und Destruktion eines Individuums manifestieren sich im dynamischen Modell durch eine entsprechende Änderung der dynamischen Extension.

<sup>1</sup> *Kind* hingegen ist doppeldeutig: Zum einen ist es eine Rolle ("das Kind von jemandem" ist fundiert, trägt aber nichts zur Identität der Person bei), zum anderen bezeichnet es genau wie *Kleinkind* bloß den Lebensabschnitt einer Person und ist damit nicht fundiert. Interessanterweise wird diese Doppeldeutigkeit im Deutschen auf der geschlechtlichen Ebene vermieden: Sohn und Tochter sind Rollen, Junge und Mädchen sind es nicht. Im Englischen ist das übrigens genauso, im Ungarischen und Chinesischen dagegen nicht [Mrva 1999].

<sup>2</sup> Die Zahl der Individuen kann endlich (und auch recht klein) sein: In der Softwaremodellierung benötigt man Individuen nur für die Darstellung von Szenarien (Kollaborationen, s. u.), die wiederum nur Beispielcharakter haben. Eine Aufzählung aller in einem Modell vorkommenden Individuen ist also durchaus realistisch.

<sup>3</sup> Die Denotation wird hier nur angeführt, um den Unterschied zwischen semantischer Interpretation und Extension deutlich zu machen. Beides sind nämlich mit einem Typsymbol assoziierte Mengen, jedoch rangieren sie über verschiedene Grundbereiche: den Grundbereich der Interpretation bzw. die Menge der Individuenbezeichner.

Neben den Spezies mit ihren Individuen sind Relationen mit ihren Tupeln (die Tupel werden im folgenden *Assoziationen* genannt) das zweite große Sprachkonstrukt der Modellierung. Ein statisches Modell umfaßt also eine Menge von Relationssymbolen  $P$  (für Prädikationen,  $R$  wird den Rollen vorbehalten), die, nach Signaturen geordnet, üblicherweise als eine Mengenfamilie

$$(P_w)_{w \in S^*}$$

eingeführt werden. Ich werde die genaue Definition von Relationen als Sprachkonstrukt jedoch hintanstellen und zunächst den Rollenbegriff behandeln.

Nach Guarino ist eine Rolle ja ein Konzept, das fundiert und nicht semantisch rigide ist. Fundierung bedeutet, daß jedes Individuum dieses Konzepts in Beziehung zu mindestens einem anderen stehen muß (sonst fällt es per definitionem nicht unter das Konzept!), ein Umstand, dem in einem (statischen) Modell dadurch Rechnung getragen wird, daß dieses Konzept an einer Relation beteiligt ist, also an einer Argumentstelle in der Deklaration dieser Relation auftaucht. Mangelnde Rigidität drückt sich darin aus, daß ein Individuum, das unter dieses Konzept fällt, nicht zu diesem Konzept gehört wie das Individuum zu seiner Spezies, durch deren Charakterisierung es seine Identität erst erlangt. Dies läßt sich wiederum so interpretieren, daß Rollen keine eigenen Individuen haben wie Spezies – sie müssen ihre Extensionen aus denen der Spezies rekrutieren.<sup>4</sup>

Ich erweitere daher die Modellierungssprache um ein Rollenkonstrukt wie folgt. Neben der Menge der Typsymbole umfasse ein Modell eine Menge von Rollensymbolen  $R$ . Typen und Rollen werden durch eine zweistellige Relation  $<_{SR} \subseteq S \times R$  in Beziehung gesetzt; die Extension dieser Relation ist Teil des statischen Modells und gibt an, aus welchen Spezies sich eine Rolle rekrutieren oder, im Rollenjargon, die Individuen welcher Spezies welche Rolle spielen können.  $Person <_{SR} Student$  beispielsweise drückt aus, daß Individuen der Spezies *Person* die Rolle *Student* spielen können (die Spezies *Person* die Rolle *Student* "füllt"). Anders als Spezies haben Rollen keine eigenen Individuen. Dennoch ist die Extension einer Rolle nicht leer (denn es gibt ja z. B. Individuen, die Student sind); vielmehr sei sie durch

$$\text{ext}(r) := \bigcup_{s <_{SR} r} \text{ext}(s)$$

definiert. Es gehören also zur Extension einer Rolle die Instanzen aller Spezies, die diese Rolle füllen. Entsprechend ließe sich nun auch die dynamische Extension einer Rolle definieren, nur dann wäre diese dynamische Extension ausschließlich von der Erzeugung und Destruktion von Individuen der Spezies bestimmt, so daß Rollen kaum eine andere Bedeutung als Generalisierungen (s. u.) hätten. Statt dessen mache ich die dynamische Extension einer Rolle von der der Relationen abhängig, wie ich gleich zeigen werde. Zur Intension einer Rolle gehört die Beschreibung aller Eigenschaften, die die Rolle charakterisieren; sie unterscheidet sich damit zunächst nicht prinzipiell von der einer Spezies. Rollen sind genau wie Spezies Typen, eben nur keine natürlichen.

Anstatt nun Relationen wie üblich auf Spezies (Typen) zu deklarieren, deklariere ich sie auf Rollen.<sup>5</sup> Zu einem statischen Modell gehört also eine Mengenfamilie von Relationssymbolen

<sup>4</sup> Natürlich ist weder die Deklaration eine Rolle an der Stelle einer Relation noch die Tatsache, daß Rollen keine eigenen Instanzen haben, zunächst ausreichend, um Guarinos Bedingungen zu erfüllen; ich werde später darauf zurückkommen.

<sup>5</sup> Diese Idee findet man bereits bei Elmasri et al. [EWH 1985] und Chu & Zhang [ChZh 1997]; keiner dieser Ansätze ist jedoch meines Wissens weiter verfolgt worden.

$$(P_w)_{w \in R^*}$$

Die  $P_w$  seien paarweise disjunkt; außerdem sei  $P_w = \emptyset$  für  $|w| < 2$  und für alle die  $w \in R^*$ , in denen ein  $r$  mehrfach vorkommt. Relationen sind also mindestens zweistellig, und innerhalb einer Relation ist jede Rolle eindeutig.<sup>6</sup>

Zu jedem  $p \in P_{r_1 \dots r_n}$  gehört nun wiederum

- eine Intension  $\text{int}(p)$ , die die Eigenschaften der Assoziationen der Relation beschreibt (die Rollen  $r_1, \dots, r_n$  an den Argumentstellen sind ein Teil davon),
- eine statische Extension  $\text{ext}(p)$ , die alle Assoziationen umfaßt, die jemals unter die Relation fallen (wenn  $\text{int}(p)$  nur aus der Angabe der Rollen besteht, ist  $\text{ext}(p) = \text{ext}(r_1) \times \dots \times \text{ext}(r_n)$ ; vgl. die Definition von *typenrecht* unten),
- eine dynamische Extension  $\text{ext}_t(p)$ , die alle Assoziationen umfaßt, die zum Zeitpunkt  $t$  unter die Relationen fallen, und
- eine Interpretation im semantischen Sinne.

Es gilt auch hier, daß stets  $\text{ext}_t(p) \subseteq \text{ext}(p)$  ist. Ob und wie  $\text{ext}(p)$  im Rahmen des statischen Modells festgelegt wird (etwa in Analogie zu den Individuen der Spezies), ist hier nicht weiter von Interesse; es gilt aber, daß alle Assoziationen *typenrecht* sein müssen, d. h., daß für ein

$$(a_1, \dots, a_n) \in \text{ext}(p) \text{ mit } a_1 \in I_{s_1}, \dots, a_n \in I_{s_n} \text{ und } p \in P_{r_1 \dots r_n}$$

$$s_1 <_{SR} r_1, \dots, s_n <_{SR} r_n$$

sein muß. Die Typeneinschränkung der Argumentstellen einer Relation erfolgt damit, anders als in der gewöhnlichen Modellierung, indirekt über Rollen.

Mit diesen Festlegungen läßt sich die dynamische Extension einer Rolle wie folgt definieren:

$$\text{ext}_t(r) := \{a \mid \exists p \in P_{\dots r \dots} : (\dots, a, \dots) \in \text{ext}_t(p)\},$$

wobei  $r$  und  $a$  in  $\dots r \dots$  und  $(\dots, a, \dots)$  an derselben Stelle stehen und die ausgelassenen Variablen alle existenzquantifiziert sein sollen. Man beachte, daß damit

$$\text{ext}_t(r) \subseteq \bigcup_{s <_{SR} r} \text{ext}_t(s)$$

ist. Es umfaßt also die dynamische Extension einer Rolle zum Zeitpunkt  $t$  all die Individuen, die in dieser Rolle an irgendeiner zu dem Zeitpunkt existierenden (also im dynamischen Extent einer Relation vertretenen) Assoziation partizipieren. Aber damit sind Guarinos beide Bedingungen genau erfüllt: Alle Instanzen einer Rolle (die Elemente der dynamischen Extension) stehen notwendig in Beziehung zu einer anderen, und keine der Instanzen verliert ihre Identität, nur weil sie eine Rolle nicht mehr spielt.

Um die Modellierungssprache, die neben Spezies (natürlichen Typen) und Relationen jetzt auch Rollen enthält, zur Basis der objektorientierten Modellierung zu machen, bedarf es noch der Möglichkeit zur Generalisierung und der damit einhergehenden Vererbung. Terminologisch vorgesehen [ISO 1987], wenn auch wenig gebräuchlich, ist, die Generalisierung (Wortstamm!) von Spezies *Genera* zu nennen (der Baum des Porphyrios

<sup>6</sup> Eine Rolle bezeichnet damit eine Stelle einer Relation eindeutig. Genau dies ist die Funktion von Rollennamen im relationalen und im ER-Modell. Auf die Fassung der Indexmenge als Potenzmenge (anstatt als Menge von Strings), die sich eigentlich aufdrängt, verzichte ich um der einfacheren Darstellung willen.

[Sowa 1992]). Dabei soll gelten, daß die Extension (statisch wie dynamisch) eines Genus genau die Vereinigung der Extensionen seiner Spezies ist.<sup>7</sup> Entsprechend kann man mehrere Rollen in einer Oberrolle zusammenfassen und die Deklaration von Relationen auch auf Oberrollen erlauben. Die Details sind nicht weiter interessant; man beachte nur, daß Genera Typen sind, deren dynamische Extensionen im Gegensatz zu der der Rollen nach unserer Definition ausschließlich von denen der Spezies bestimmt werden, von denen sie abstrahieren. Genera sind also im Gegensatz zu Rollen semantisch rigide (und nicht fundiert); es sind natürliche Typen.

Es bleibt eine Bemerkung zur Intension von Spezies und Rollen: In der objektorientierten Softwaremodellierung zählen dazu Methoden und Attribute. Man beachte, daß diese Eigenschaften nicht nur von den Genera zu ihren Spezies und von den Rollen auf ihre Unterrollen, sondern auch von den Rollen zu ihren rollenfüllenden Genera und/oder Spezies vererbt werden. Dies folgt aus der Mengeninklusion der Extensionen. Entsprechend wird die Eigenschaft, an einer Relation zu partizipieren, von allen Rollen auf ihre Unterrollen und wiederum auf die rollenfüllenden Genera und/oder Spezies vererbt.

## 4 Übertragung auf die objektorientierte Softwaremodellierung

Derzeit konkurrieren viele Sprachen zur objektorientierten Softwaremodellierung um die Gunst der Softwaretechniker, aber am lautesten diskutiert wird wohl UML. In UML gibt es zwei verschiedene Rollenbegriffe, die zunächst nur wenig miteinander zu tun zu haben scheinen: die Rolle als das Ende einer Assoziation und die Rolle als ein Akteur in einer Kollaboration. Mit der ersten Auffassung folgt UML OMT [RBP+ 1991] und damit der Diktion des ER-Modells [Chen 1976] (und schließlich auch des relationalen Modells [Codd 1970]), wonach Rollen Namen sind, die die Stellen einer Relation eindeutig bezeichnen.<sup>8</sup>

Bei den Kollaborationen handelt es sich um ein Modellierungskonstrukt, das der Darstellung der Interaktion mehrerer Instanzen zur gemeinsamen Erfüllung einer bestimmten Aufgabe dient. Dabei können Instanzen verschiedener Klassen in derselben Rolle und Instanzen derselben Klasse in verschiedenen Rollen an einer Kollaboration teilhaben; es kann sogar ein und dieselbe Instanz verschiedene Rollen spielen. Solchermaßen ist in einem Klassendiagramm nicht darstellbar. Mehr noch: Es offenbart sich hier genau die Problematik, die schon Bachman und Daya [BaDa 1977] erkannten: Das Konzept der Klasse (oder des Typs), dem jede Instanz eindeutig (bis auf Generalisierung) zugeordnet ist, taugt nicht für die Modellierung von Rollen. Die Konsequenz ist, daß Klassen in einem Kollaborationsdiagramm nichts zu suchen haben.<sup>9</sup>

UML trägt dem Rechnung, indem es auf der sog. Spezifikationsebene der Kollaborationsdiagramme Classifier roles einführt und diese über Association roles und deren Association end roles verbindet. Dabei sind diese "Rollen" im wesentlichen auf die Anforderung

<sup>7</sup> Daß Genera selbst keine eigenen Instanzen haben, ist keine praktische Einschränkung und kann als gute Praxis aufgefaßt werden [Stein 2000a].

<sup>8</sup> Genaugenommen ist in UML 1.3 der Rollenname das *Name*-Attribut eines *AssociationEnd*, dem Sprachelement, dem die Rolleninformationen zugeschrieben werden [OMG 1999, § 2.5.2]. Der Begriff *Rolle* wird dafür zunehmend seltener verwendet, wohl um Verwechslung mit den Rollen in Kollaborationen zu vermeiden.

<sup>9</sup> Auf dieser Einsicht (sowie auf Kollaborationen allgemein) basieren unter anderem die Rollenmodelle von OORAM [RWL 1996] und CATALYSIS [D'SWi 1998], die hier einen maßgeblichen Einfluß auf die Definition von UML hatten.

rungen des jeweiligen Verwendungszwecks innerhalb der Kollaboration gerichtete Sichten oder Projektionen von Klassen (und was in UML sonst noch so unter Classifier fällt) bzw. Assoziationen und deren Assoziationsenden. Dadurch, daß dieselbe Klasse verschiedene Projektionen haben kann (durch verschiedene Classifier roles repräsentiert) und dadurch, daß eine Classifier role die Projektion mehrerer Klassen sein kann, liegt mit einem Kollaborationsdiagramm auf Spezifikationsebene praktisch eine Ad-hoc-Rollendeklaration im Sinne von Abschnitt 3 vor. Abb. 1 zeigt ein solches Kollaborationsdiagramm auf Spezifikationsebene; die Klasse *Person* kommt darin in zwei Classifier roles vor, nämlich als *Teacher* und *Student*.

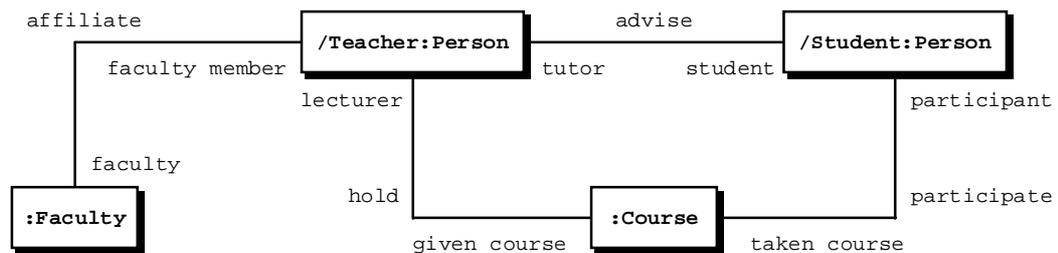


Abb. 1: Kollaborationsdiagramm in UML (aus [OMG 1999]; ergänzt um Relationsnamen). Man beachte den Namenskonflikt zwischen den Rollennamen an den Assoziationsenden und den Namen der Classifier roles.

Auf der sog. Instanzebene wird die Spezifikation einer Kollaboration mit Instanzen bevölkert: Anstelle der Classifier roles stehen Objekte und an der der Association roles Links. Dabei können die Objekte anonym bleiben (sie werden dann nur durch einen Rollen- und/oder Klassennamen spezifiziert); die Links sind es sowieso. Indem man in Abb. 1 die Rollennamen innerhalb der Kästen unterstreicht, wird daraus ein Kollaborationsdiagramm auf Instanzebene mit lauter anonymen Objekten, das allerdings nur eins von vielen möglichen ist – eins, das ausdrückt, daß derselbe Lehrer denselben Studenten sowohl unterrichtet als auch betreut.

Will man die Relationen des Modells aus Abb. 1 auf die oben skizzierten Modellierungssprache zurückführen, dann werden, die entsprechenden Rollendeklarationen vorausgesetzt, diese per

$$affiliate \in P_{faculty\ member\ faculty}$$

$$advise \in P_{tutor\ student}$$

$$hold \in P_{lecturer\ given\ course}$$

$$participate \in P_{participant\ taken\ course}$$

deklariert. Dabei wird jeder Rollename am Ende einer Assoziation als Rolle interpretiert. Wenn man dann noch die Classifier roles *Teacher* und *Student* als Unterrollen per

$$Teacher \leq_{RR} faculty\ member \quad Teacher \leq_{RR} tutor \quad Teacher \leq_{RR} lecturer$$

und

$$Student \leq_{RR} student \quad Student \leq_{RR} participant$$

hinzufügt (wobei  $\leq_{RR}$  die Unterrollenrelation bezeichne), kann man zudem auf elegante Art die Namenskonflikte zwischen den (Rollen-)Namen, die das Ende einer Association role bezeichnen, und denen der Classifier roles vermeiden; man muß nur die Relationsdeklarationen wie folgt überladen:

$affiliate \in P_{Teacher-faculty}$   
 $advise \in P_{Teacher-Student}$   
 $hold \in P_{Teacher-given\ course}$   
 $participate \in P_{Student-taken\ course}$

Die Rollen werden dann durch Typen (Spezies) per

$Person <_{SR} Teacher$   
 $Person <_{SR} Student$   
 $Faculty <_{SR} faculty$   
 $Course <_{SR} given\ course \quad Course <_{SR} taken\ course$

entsprechend Abb. 1 gefüllt. Man beachte, daß die Deklaration der Unterrollen ausdrückt, daß *Teacher* alle Eigenschaften von *tutor*, *faculty member* und *lecturer* vereint; entsprechendes gilt für *Student*. Dies scheint sinnvoll: Jede Relation fordert im Rahmen der Kollaboration gewisse Eigenschaften von seinen Rollen, und dort, wo später ein Individuum mehrere Rollen spielen soll, werden die Intensionen dieser Rollen vereint. Wenn Abb. 1 jedoch andeuten soll, daß *tutor*, *faculty member* und *lecturer* bzw. *student* und *participant* jeweils Rollen zweier verschiedener Personen sind (ausgedrückt durch die separaten Classifier roles *Teacher* und *Student*), dann geben das formal weder die UML-Syntax noch die obigen Deklarationen her – um Identität oder Verschiedenheit von Individuen auszudrücken, muß man schon auf die Instanzebene gehen und ein Kollaborationsdiagramm auf Instanzebene wie oben angedeutet erstellen.

## 5 Eine Umgestaltung der UML

Die Fassung des Rollenkonzeptes in UML ist derzeit (Version 1.3) in keinem guten Zustand [D'Sou 1999]. Eine nach formalen Gesichtspunkten sinnvolle Umgestaltung der UML, die das hier definierte Rollenkonzept gleichberechtigt neben die Klassen stellt, könnte etwa die folgenden Revisionen umfassen:

1. Das Klassendiagramm beschränkt sich auf

- die Spezifikation der Klassenhierarchie,
- die Spezifikation der Attribute, die nicht als Relationen (“Assoziationen” im UML-Jargon) dargestellt werden,
- die Spezifikation der Methoden und
- die Angabe der Rollen, die eine Klasse füllen kann.

Als Notation für die Rollen einer Klasse eignet sich die Lollipop-Notation, die in UML für Interfaces vorgesehen ist [FoSc 1997]. Da Rollen selbst wieder hierarchisch geordnet sein und verschiedene Klassen dieselben Rollen füllen können, ist zu überlegen, ob die Rollen- und die Klassenhierarchie gemeinsam oder in verschiedenen Diagrammen dargestellt werden sollen; denkbar ist beides. Ausdrücklich ausgeschlossen vom Klassendiagramm sind Relationen (“Assoziationen”), die ja in UML mit der Generalisierung gemeinsam in einem Diagramm (dazu auch noch beide durch Verbindungslinien) dargestellt werden. Formal ist diese Trennung zu begrüßen, da die Generalisierung (genau wie die Relation  $<_{SR}$ , das Rollenfüllen) eine Relation zweiter Ordnung ist und damit nicht in den Reigen der benutzerdeklarierten gehört.

2. Interfaces werden durch Rollen ersetzt. Die Interfaces von UML sind die Interfaces von JAVA, nicht mehr und nicht weniger. Damit werden aber die konzeptuellen Modellierungsmöglichkeiten, die hinter einer zu der der Klassen orthogonal deklarierten Hierarchie von Typen stecken, nicht annähernd ausgereizt – es fehlt dazu die Struktur, die vorgibt, wann Interfaces zu verwenden sind und wann Klassen.<sup>10</sup> Eine solche Struktur wird durch den oben definierten Rollenbegriff gegeben und durch die nachfolgenden Änderungsvorschläge 3 und 4 auf UML übertragen.
3. Es wird ein zusätzlicher Diagrammtyp eingesetzt: das Relations- (oder Assoziations-) Diagramm. Das Relationsdiagramm übernimmt zum einen die (benutzerdeklarierten) Relationen aus dem bisherigen Klassendiagramm und löst zum anderen das Kollaborationsdiagramm auf der Spezifikationsebene ab. Anders als im Klassendiagramm sind aber die Stellen der Relationen mit Rollen (und nicht wie bisher mit Klassen) besetzt; ein Relationsdiagramm spiegelt also die Relationsdeklarationen aus der oben skizzierten Modellierungssprache wider<sup>11</sup>. Ein solches Diagramm ist in Abb. 2 (a) dargestellt.

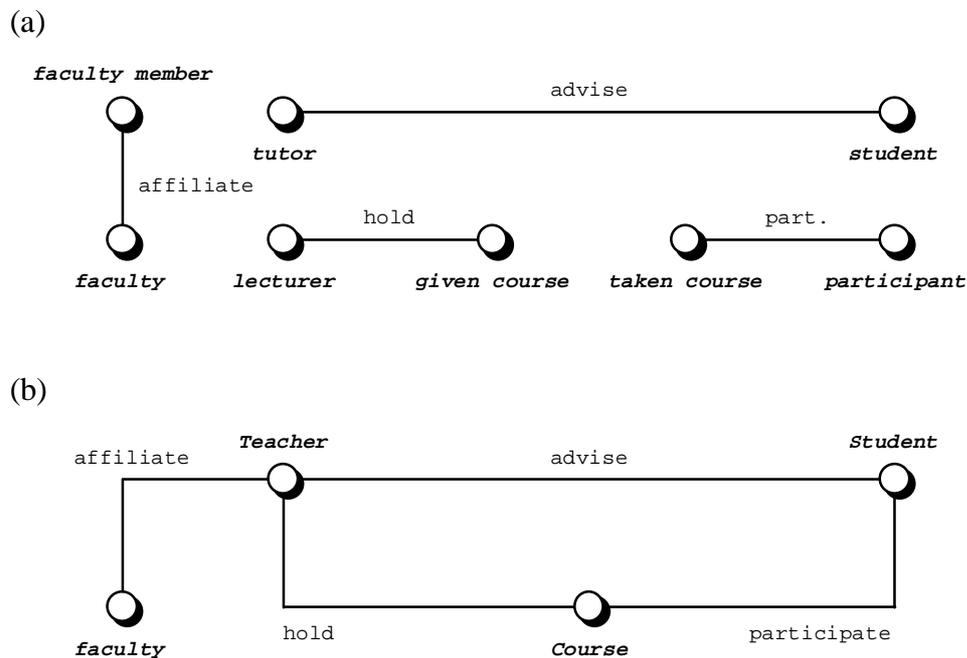


Abb. 2: Relationsdiagramme

- (a) Relationen und ihre Rollen; Teile des Diagramms sind nicht miteinander verbunden  
 (b) Ableitung aus (a), die durch Einsetzen der Unterrollen Teacher, Student und Course entstanden ist; man beachte die Analogie zu Abb. 1.

Wie man in Abb. 2 sieht, sind Teile des Relationsdiagramms zunächst unverbunden. Es drängt sich auf, die Verbindung der Teile durch das Eintragen der rollenfüllenden

<sup>10</sup> Das, was UML vorgibt, ist, daß Interfaces anders als Klassen keine Implementierung mit sich bringen. Dieses eher technische Detail trägt jedoch kaum etwas zur konzeptuellen Modellierung bei. Daß Interfaces keine Attribute haben und nur an der Zielstelle einer gerichteten Assoziation auftreten dürfen, ist wohl eher eine Verbeugung vor JAVA als wohl durchdacht. Das Rollenkonzept zumindest wird damit nur ungenügend unterstützt: Die Rolle *Student* z. B. steht in bidirektionaler Verbindung zu *Studiengang*; dies der Klasse *Person* zuzuschreiben, die die Rolle füllt, wird der Realität nicht gerecht [Stein 2000b].

<sup>11</sup> Dabei ist es durchaus denkbar (um eine Inflation von Namen zu vermeiden), daß einzelne Rollen den Namen der Klasse tragen, die sie füllt. Dies ist dann so zu verstehen, daß die Rolle die Klasse vollständig charakterisiert. *Course* aus dem Modell zu Abb. 2 ist ein Beispiel für so einen Fall.

Klassen herzustellen. Zwei Dinge sprechen dagegen: Erstens ist es durchaus denkbar, daß manche Rollen von mehreren Klassen gefüllt werden, die nicht über eine gemeinsame Oberklasse zusammengeführt werden (das ist gerade einer der Gründe für die Einführung eines Rollenkonzepts in die Modellierung [Stein 1999]), so daß das Diagramm schnell unübersichtlich wird. Zum anderen werden im gewählten Beispiel gleich fünf Rollen von einer Klasse (nämlich Person) gefüllt, so daß das Diagramm durch das Hinzufügen dieser Klasse kaum Struktur gewinnt. Statt dessen ist es sinnvoll, die Rollen über gemeinsame Unterrollen zu verschmelzen, wie in Abb. 2 (b) zu sehen ist; es entspricht dies übrigens dem Überladen der Relationsdeklarationen aus dem vorigen Abschnitt.

4. Die Darstellung der Kollaborationen erfolgt jetzt ausschließlich auf der Ebene der Instanzen, und zwar mit Hilfe eines abgewandelten Kollaborationsdiagramms. Grundlage eines Kollaborationsdiagramms ist ein Szenario, wie es in Abb. 3 dargestellt ist; ein solches Szenario setzt ausschließlich Instanzen miteinander in Verbindung und muß selbst stets Instanz eines oder mehrerer Relationsdiagramme sein. Ein Kollaborationsdiagramm wird daraus, indem die in UML verwendeten Accessoires zur Darstellung der Sequenz des Nachrichtenaustauschs hinzugefügt werden.

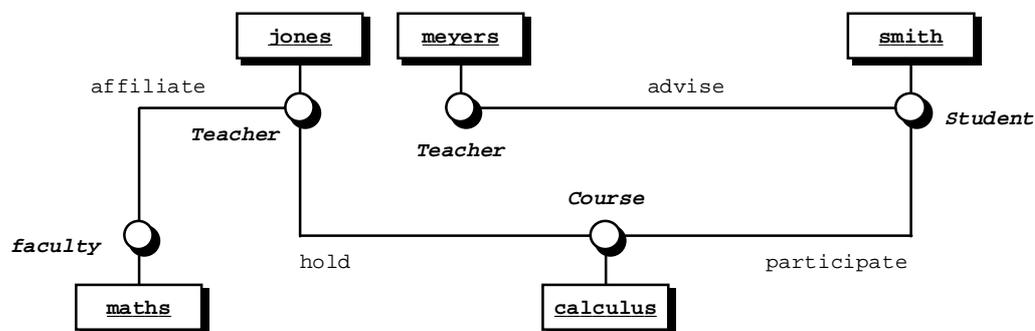


Abb. 3: Szenario mit Instanzen in ihren Rollen als Grundlage eines Kollaborationsdiagramms.

Szenarien wie das in Abb. 3 dienen aber nicht nur als Grundlage der Spezifikation von Kollaborationen, sie können durchaus auch einen eigenständigen Beitrag zur Modellierung leisten. Während z. B. die Relationsdiagramme aus Abb. 2 noch zulassen, daß dieselbe Person sowohl Lehrer als auch Student ist, legt das Kollaborationsdiagramm aus Abb. 3 zumindest nahe, daß sich eine Person nicht selbst unterrichtet. Es wird also sinnvoll sein, Relationsdiagramme zur näheren Bestimmung der Relationsdeklarationen hier und da durch solche Szenarien zu ergänzen.

Man beachte, daß sich alle vorgeschlagenen Diagrammtypen unmittelbar auf die in Abschnitt 3 vorgestellte formale Modellierungssprache zurückführen lassen. Damit wird nicht nur die automatische Prüfung der Konsistenz der verschiedenen Diagramme eines Modells, sondern auch die aktive Unterstützung des Modellierungsprozesses durch ein Modellierungswerkzeug, das auf dem zur Modellierungssprache gehörenden Metamodell basiert, möglich. Wie dieses Metamodell aussieht und welche Vorzüge es gegenüber dem der aktuellen UML-Spezifikation hat, ist in [Stein 2000b] nachzulesen.

## 6 Schluß

Ziel meiner Arbeit ist es, die klassische Dichotomie Typ/Relation in der Softwaremodellierung zu einer akzeptierten Trias Typ-Rolle-Relation aufzuweiten. Motivation, dies

zu tun, so scheint es, ist reichlich gegeben; ich verweise dazu auf die aktuelle Literatur zur objektorientierten Softwareentwicklung, in der die Verwendung von Interfaces und Delegation anstelle von Klassen und Vererbung zur Erlangung der sog. Plugability gepredigt wird (z. B. [CoMa 1999]). Die vorgeschlagene Umgestaltung von UML fällt dabei vergleichsweise moderat aus; wesentliche Ansätze des Rollenkonzepts sind bereits vorhanden und können mit kleinen Anpassungen übernommen werden [Stein 2000b]. So besteht mein Beitrag für die vollwertige Integration des Rollenkonzepts in die objektorientierte Modellierung vor allem aus zwei Dingen: einem vielen Betrachtungswinkeln standhaltenden theoretischen Unterbau sowie einer praktikablen Neustrukturierung der grafischen Notation.

## 7 Danksagung

Ich danke Michael Mrva für seine wertvollen Einsichten und Hinweise zum Thema Rollen und den anonymen Gutachtern für ihre Anregungen.

## Literatur

- [BaDa 1977] CW Bachman, M Daya “The role concept in data models” in: *Proceedings of the 3rd International Conference on Very Large Databases* (1977) 464–476.
- [Booc 1994] G Booch *Object-Oriented Analysis and Design with Applications* (Addison-Wesley, Menlo Park 1994).
- [Bühl 1934] K Bühler *Sprachtheorie: Die Darstellungsfunktion der Sprache* (Gustav Fischer, Jena 1934).
- [Chen 1976] PP Chen “The entity-relationship model: Towards a unified view of data” *ACM Transactions on Database Systems* 1:1 (1976) 9–36.
- [ChZh 1997] WW Chu, G Zhang “Associations and roles in object-oriented modeling” in: DW Embley, RC Goldstein (eds) *Proceedings of the 16<sup>th</sup> International Conference on Conceptual Modeling: ER '97* (Springer-Verlag, Berlin 1997) 257–270.
- [CoMa 1999] P Coad, M Mayfield *JAVA Design: Building Better Apps and Applets* 2. Ausgabe (Yourdon Press, Upper Saddle River 1999).
- [Codd 1979] EF Codd “A relational model of data for large shared data banks” *Communications of the ACM* 13:6 (1970) 377–387.
- [D'SWi 1998] DF D'Souza, AC Wills *Objects, Components and Frameworks with UML* (Addison-Wesley, Reading 1998).
- [D'Sou 1999] DF D'Souza *persönliche Kommunikation* (1999).
- [Eco 1994] U Eco *Die Suche nach der vollkommenen Sprache* (C. H. Beck, München 1994).
- [EWH 1985] R Elmasri, J Weeldreyer, A Hevner “The category concept: an extension to the entity relationship model” *Data & Knowledge Engineering* 1:1 (1985) 75–116.

- [Falk 1976] E Falkenberg “Concepts for modelling information” in: GM Nijssen (ed) *Proceedings of the IFIP Conference on Modelling in Data Base Management Systems* (North-Holland, Amsterdam 1976) 95–109.
- [Fill 1987] CJ Fillmore “Types of lexical information” Auszüge abgedruckt in: R Dirven, G Radden *Fillmore’s Case Grammar: A Reader* (Julius Groos Verlag, Heidelberg 1987).
- [FoSc 1997] M Fowler, K Scott *UML Distilled* (Addison-Wesley, Reading 1997).
- [Guar 1992] N Guarino “Concepts, attributes and arbitrary relations” *Data & Knowledge Engineering* 8 (1992) 249–261.
- [Huss 1901] E Husserl *Logische Untersuchungen* 2. Band, 1. Teil, 3. Auflage (Max Niemeyer, Halle a. d. Saale 1922).
- [ISO 1987] ISO 704 *Principles and Methods of Terminology* (International Organization for Standardization, 1987).
- [Lodw 1647] F Lodwick *A Common Writing* abgedruckt in: V Salmon *The Works of Francis Lodwick* (Longman, London 1972).
- [Mrva 1999] M Mrva *persönliche Kommunikation* (1999).
- [NiHa 1989] GM Nijssen, TA Halpin *Conceptual Schema and Relational Database Design: a Fact-Oriented Approach* (Prentice-Hall, 1989).
- [OMG 1999] OMG *Unified Modeling Language Specification V 1.3* (<http://www.omg.org>).
- [RWL 1996] T Reenskaug, P Wold, OA Lehne *Working with Objects – The OOram Software Engineering Method* (Manning, Greenwich 1996).
- [RBP+ 1991] J Rumbaugh et al. *Object-Oriented Modeling and Design* (Prentice Hall, Englewood Cliffs 1991).
- [Sowa 1984] JF Sowa “Semantic networks” in: SC Shapiro (ed) *Encyclopedia of Artificial Intelligence* 2<sup>nd</sup> edition (John Wiley, New York 1992) 1493–1511.
- [Stei 1999] F Steimann “On the role of roles in object-oriented and conceptual modelling” eingereicht bei *Data & Knowledge Engineering* (1999).
- [Stei 2000a] F Steimann “Abstract class hierarchies, factories, and stable designs” *Communications of the ACM* 43 (2000) in press.
- [Stei 2000b] F Steimann “Role = Interface: a merger of concepts” eingereicht bei *Journal of Object-Oriented Programming* (2000).
- [Tesn 1965] L Tesnière *Éléments de syntaxe structurale* 2nd edition (Librairie C Klincksieck, Paris 1965).
- [WdS 1995] R Wieringa, W de Jonge, P Spruit “Using dynamic classes and role classes to model object migration” *Theory and Practice of Object Systems* 1:1 (1995) 61–83.

Anschrift des Autors:

Universität Hannover, Institut für Technische Informatik, Rechnergestützte Wissensverarbeitung (KBS),  
 Appelstraße 4, 30167 Hannover; [steimann@acm.org](mailto:steimann@acm.org).