Namensrecht und objektorientierte Analyse mit UML

Fragen über Fragen

Burkhardt Renz

Fachhochschule Gießen-Friedberg, Fachbereich MNI,
Wiesenstr. 14, D-35390 Gießen
Burkhardt.Renz@mni.fh-giessen.de

Rev 2.5 – 31. Januar 2005

Zusammenfassung

An einem Beispiel – dem des Namensrechts – werden Fragen zur objektorientierten Analyse aufgeworfen. Aus diesen Fragen ergibt sich ein Plädoyer dafür, für die Beschreibung eines Anwendungsgebiets die jeweils *geeignete* Methode zu verwenden.

Einleitung

Die Befürworter der objektorientierten Analyse sind der Auffassung, dass die Objektorientierung nicht nur eine Idee der Konstruktion von *Code*, von Softwarestrukturen ist, sondern es in natürlicher Weise erlaubt, die Sache selbst, die reale Welt, zu erfassen. Ein Zitat mag genügen:

> „Many objects are there just for the picking. They directly model objects of the physical reality to which the software applies. One of the particular strengths of object technology is indeed its power as a modeling tool, using software object types (classes) to model physical object types, and the method's inter-object-type relations (client, inheritance) to model the relations that exist between physical object types, such as aggregation and specialization.“ ([5, S. 117]

Nun ist es in der Tat so, dass bei einer großen Klasse von Problemen, namentlich den Informationsproblemen, für die Konstruktion von Software Modellbereiche benötigt werden, die ein Analogon zur realen Welt darstellen,

und mit denen die Software agiert, sie verändert, speichert usw. Das Zitat behauptet, dass solche Modelle gewissermaßen „auf der Straße liegen“ und nur aufgepickt werden müssen. In den Lehrbüchern werden oft nur einfache Ausschnitte realer Probleme besprochen, meist sind es simple Situationen, an denen die behauptete Entsprechung demonstriert wird.

Da ist das Erstellen eines Domänenmodells des deutschen Namensrechts schon eher eine echte Aufgabe. Es ist ja selbst für Experten des Gebiets unter Umständen nicht einfach zu sagen, welche Namensgebung erlaubt ist, welche Namen wie und warum zustandegekommen sind. In diesem Artikel wird eine Musterlösung für diese Aufgabe diskutiert, die in den Übungen zur Vorlesung „Softwaretechnik III“ an der Fachhochschule Gießen-Friedberg entstanden ist.

Das Beispiel eignet sich dazu, Fragen zu stellen. Fragen, die über das Beispiel hinausgehen und auch das einleitende Zitat eines Advokaten der Objektorientierung in einem anderen Licht erscheinen lassen: Wiewohl objektorientierte Codestrukturen für eine Software im Bereich des Personenstandswesens fraglos verwendet werden können, wirft die Musterlösung für ein Domänenmodell die Frage auf, ob objektorientierte Analyse für ein solches Problem geeignet ist, und ob die Notation mit der Unified Modeling Language UML für die Darstellung und Erörterung der Analyseergebnisse wirklich taugt.

Vorweg sei bemerkt:

- Jede Analyse und Kritik hängt natürlich vom Maßstab ab, von dem Anspruch, der angelegt wird. Ich expliziere diesen Maßstab im ersten Abschnitt des Artikels. Wenn mit der Musterlösung ein anderer Anspruch verbunden wird, dann ist die Kritik womöglich gegenstandslos.

- Von Kleinigkeiten abgesehen, sehe ich wenig Möglichkeiten das Modell mit den Mitteln der Objektorientierung und der UML wirklich zu verbessern. Sicherlich kann man mit der Object Constraint Language das Modell präzisieren – aber viele der Fragen werden bleiben. Anliegen dieses Artikels ist es eher, Zweifel zu streuen, ob die objektorientierte Analyse für *diesen Typ* Problem das geeignete Mittel ist.

- Die Kritik mag manchmal so erscheinen, aber sie ist es nicht: spitzfindig. Wer den Prozess der Softwareentwicklung kennt, wird wissen, dass im Laufe der Entwicklung alle der in diesem Artikel gestellten Fragen auftreten. Je später, desto schlechter oft – weil ihre Klärung drastische Eingriffe in bereits entwickelte Teile der Software erfordern wird.

1 Charakteristik der Domäne und Aufgaben der Analyse

1.1 Zum Begriff des Modells

Folgt man der UML und dem Rational Unified Process (RUP) ist die Aufgabe der Analyse die Erstellung eines Domänenmodells:

> „**domain** — An area of knowledge or activity characterized by a set of concepts and terminology understood by practitioners in that area.“ ([6, Kap. 4])
>
> „**domain model** — A domain model captures the most important types of objects in the context of the domain. The domain objects represent the entities that exist or events that transpire in the environment in which the system works.“ ([8, Glossary])
>
> „**analysis** — The phase of the system development process whose primary purpose is to formulate a model of the problem domain that is independent of implementation considerations. Analysis focuses on what to do; design focuses on how to do it.“ ([6, Kap. 4])

Das klingt ganz plausibel, doch schauen wir etwas näher hin: Was ist mit *Modell* genau gemeint?

Man kann unterscheiden zwischen *analytischen* und *analogen* Modellen:

Ein analytisches Modell abstrahiert eine Sache auf ihre wesentlichen Merkmale und fasst sie in Gesetzmäßigkeiten. So verwenden Physiker das Modell des Massenpunkts (der keine Ausdehnung hat) zur Erklärung und Beschreibung mechanischer Phänomene. Das Modell fasst das Wesentliche ins Auge, es dient der Beschreibung, mit ihm kann man die Phänomene und das Verhalten eines Bereichs analysieren. Oft werden analytische Modelle mathematisch formuliert. Man kann an ein analytisches Modell die Frage stellen, ob es zutrifft oder nicht.

Ein analoges Modell ist keine Erklärung einer Sache, sondern ihre Nachbildung in einem anderen Medium. Eine Modelleisenbahn ist eine Eisenbahn, aber keine wirkliche Eisenbahn, sondern verkleinert, mit anderen Materialien gebaut, usw. Bei einem analogen Modell kann man nicht fragen, ob es zutrifft, sondern nur, inwiefern es mit der modellierten Sache übereinstimmt. Welche Übereinstimmung man anstrebt, hängt davon ab, was man mit dem analogen Modell machen möchte, welchen Zweck man mit ihm verfolgt.

Um den Unterschied festzuhalten, werde ich im Folgenden von einer *Beschreibung* sprechen, wenn die Analyse der Sache selbst gemeint ist, und von einem *Modell*, wenn es sich um ein analoges Modell handelt. Ich folge darin [4].

Dieser Unterschied wird in obiger Zielvorgabe für die Analyse nicht gemacht, obwohl er wichtig ist: Wollen wir ein (analytisches) Modell des Na-

mensrechts erstellen, das wahre Aussagen über das Namensrecht enthält, aus denen man weitere Aussagen herleiten kann usw.? Oder wollen wir ein (analoges) Modell, in dem bestimmte Aspekte des Namensrechts als Objekte im Speicher eines Computers simuliert werden? Tatsächlich wird man für die Entwicklung einer Software für das Gebiet des Personenstandswesens *beide* „Modelle“ benötigen.

In vielen Texten zur objektorientierten Analyse wird von Modellen gesprochen, ohne klar zu definieren, welcher Begriff von „Modell“ denn gemeint ist. Es ist unklar, ob der genannte Unterschied bewusst ist.[1]

Man muss allerdings den Eindruck gewinnen, dass mit dem Begriff Domänenmodell *beides* in *einem* Modell gemeint ist. Implizit wird die Behauptung aufgestellt, dass im Grunde sich die „Objekte“ der wirklichen Welt und die „Objekte“ im Speicher des Computers entsprechen, jedenfalls in Bezug auf das zu konstruierende System. So argumentiert ja auch Bertrand Meyer im Zitat in der Einleitung. Wie kann man allerdings feststellen, welches Aspekte der Wirklichkeit das analoge Modell abbildet (und worin es abweicht), wenn man gar kein analytisches Modell zum vergleichenden Maßstab zur Verfügung hat?

Die Musterlösung zum Namensrecht soll darauf abgeklopft werden, inwiefern sie das Versprechen dieser Entsprechung einlöst. Doch machen wir uns erst ein paar Gedanken darüber, wodurch sich der *Problem*bereich Namensrecht auszeichnet.

1.2 Charakteristik der Domäne

Betrachten wir zunächst die Charakteristik der Domäne in der wirklichen Welt, die des Namensrechts, im Anschluss dann die eines denkbaren analogen Modells dafür. Die Begrifflichkeit stammt von Michael Jackson, siehe [4].

Das Namensrecht ist ein *gegebener* Bereich, er wird nicht entworfen. Für ein Softwaresystem, das sich mit dem Namensrecht befasst, kann der Bereich nicht verändert oder angepasst werden. Er ist *autonom* und *kausal*, denn er bringt Phänomene unabhängig von einer Software hervor und er verursacht Änderungen in der Realität, die Namensgebung wird durch das Namensrecht ja tatsächlich bestimmt.

Gleichzeitig kann man die Phänomene des Bereichs als *symbolische* sehen. Der Name einer Person ist die Beziehung einer Entität zu einem Wert, dieser Zustand kann wechseln durch Namensänderung. Und der Bereich umfasst sogar Phänomene auf einer Metaebene, nämlich die Regeln selbst, in denen das Namensrecht ausformuliert ist.

Hinzukommt eine zusätzliche Eigenart des Bereichs, er hat einen *informellen* Aspekt (*informal flavour*): die Gesetze werden ausgelegt, sie sind

[1] In der Einleitung seines Buches über Analysis Patterns kommt Martin Fowler auf den Begriff des Modells zu sprechen und verweist sogar auf Beispiele aus der Physik. Diese Beispiele sind eindeutig analytische Modelle, die er gleichsetzt mit den Modellen in seinem Buch, die wiederum offensichtlich analoge Modelle sind. [2]

in ihrer Auslegung umstritten, sie können auf eine gesellschaftliche Realität stoßen, für die sie gar nicht gedacht waren usw.

Ein analoges Modell des Namensrecht ist seiner Natur nach ein *lexikalischer* Bereich. Es ist ein *entworfener Modellbereich*. Für die Art des Entwurfs sind zwei Gesichtspunkte von Bedeutung: die Korrespondenz zum Bereich des Namensrechts, also zur Sache selbst und die beabsichtigte Verwendung des Modellbereichs.

Man kann sich unterschiedliche Modelle ja nach Art des Einsatzes in einem Softwaresystem vorstellen:

- So kann man sich für den Aufbau der Namen und der Namensbestandteile interessieren. Ein solches Modell wird Namensbestandteile wie Ehenamen, vorangestellter Geburtsname, Aliasname, Künstlername u.ä. unterscheiden, ganz unabhängig davon, wie diese Namen rechtlich zu bewerten sind.

- Eine Software könnte sich für das Zustandekommen von Namen unter rechtlichen Gesichtspunkte interessieren und die Namensgebung dokumentieren.

- Eine Software könnte die Regeln verwenden wollen, nach denen die Namensgebung und -bestimmung geschieht; etwa zur Überprüfung standesamtlicher Vorgänge oder zur Steuerung der Bearbeitung von Personenstandsfällen. In diesem Fall wird nicht nur ein Modell der Namen, sondern auch der Regeln des Namensrechts benötigt.

Wichtig ist: für das Erstellen eines solchen (analogen) Modells braucht man als Grundlage in jedem Fall eine *Beschreibung* des Bereichs Namensrecht. (Wobei je nach Art des angestrebten analogen Modells eventuell nur Ausschnitte der Beschreibung benötigt werden.)

1.3 Aufgabe und Ziel der Analyse

Folgt man diesen Überlegungen, so hat die Analyse *zwei* Aufgaben:

1. Beschreibung des Namensrechts

2. Bilden eines geeigneten Modellbereichs

Für die Beschreibung des Namensrechts sind die gesetzlichen Bestimmungen die Grundlage. Leider kann man sie nicht einfach so, wie sie sind, als Beschreibung verwenden. Denn die Bestimmungen sind auf verschiedene Gesetze verstreut, Gerichtsurteile müssen berücksichtigt werden – und für eine brauchbare Beschreibung sollten die Bestimmungen systematisch dargestellt sein. Diese Systematisierung muss auch die Punkte aufdecken, bei denen Unklarheiten in den gesetzlichen Bestimmungen zu finden sind. Denn solche Unklarheiten müssen ja beim Bilden des analogen Modells unter Umständen berücksichtigt werden.

Eine Beschreibung des Namensrechts sollte Fragen der folgenden Art beantworten können: Kann es Namen mit zwei Bindestrichen geben? Wenn ja, wie können solche Namen zustandekommen? Welche früheren Namen kann eine Person einem Ehenamen voranstellen oder anfügen? Können zwei Geschwister aus einer Ehe zwei verschiedene Namen haben? Können zwei verheiratete Partner verschiedene Namen haben? Kann ein nicht eheliches Kind den Namen des Vaters als Geburtsnamen bekommen? Usw. usf.

Die Beschreibung des Problembereichs muss in einer Sprache erfolgen, die für die Domänenexperten gut verständlich ist. In manchen Bereichen findet man als Analytiker eine solche Sprache bereits vor (etwa bei den eingangs erwähnten Beispielen aus der Physik), in unserem Fall muss man die Beschreibung erstellen. Das Kriterium der Verständlichkeit der Beschreibung ist sehr wichtig: nur wenn es erfüllt ist, werden gravierende Missverständnisse über die Sache vermieden. Im Gegensatz dazu braucht ein analoges Modell diesem Kriterium nicht zu genügen und kann es wahrscheinlich auch nicht mehr. In ihm können Formalismen der Softwaretechnik verwendet werden, die den *begriffenen* Problembereich in ein für eine Software geeignetes Modell *transformieren*.

Ein solcher Modellbereich kann natürlich mit objektorientierten Techniken dargestellt werden: Die Objekte simulieren dann die wirklichen Dinge im Computer. Je nach Anforderungen an die Software wird sich der Modellbereich unterschiedlich darstellen. Oben wurden denkbare Ausprägungen von Software im Bereich Personenstandswesen angedeutet; sie brauchen unterschiedliche Modelle. In dieser zweiten Aufgabe hat die Analyse also das Ziel, ein für den Zweck der Software geeignetes Modell zu bilden und die dafür notwendige Korrespondenz zur Beschreibung des Fachgebiets zu garantieren.

1.4 Maßstab der folgenden Diskussion

Aus den bisherigen Überlegungen ergibt sich der Maßstab der folgenden Diskussion der Musterlösung zum deutschen Namensrecht.

Die objektorientierte Analyse verwischt den Unterschied zwischen Beschreibung und Modell, sie behauptet aber im Grunde *beides* in einem Zuge zu erfassen. Also werden wir in den folgenden Abschnitten auch beides *fragen*:

- Welche Aussagen über das deutsche Namensrecht lassen sich durch das Modell bestätigen oder widerlegen?
- In wieweit korrespondiert das Modell zur Sache selbst? Welche Aspekte werden getreu, welche ungenau oder gar nicht wiedergegeben?

Die Frage, ob das Modell den Anforderungen an die zu konstruierende Software gerecht wird, können wir nicht stellen, weil keine solche Anforderungen vorgegeben waren.

2 Analyse des Domänenmodells zum deutschen Namensrecht

In diesem Abschnitt beziehe ich mich auf die Musterlösung zum deutschen Namensrecht in Abb. 1.

Um die Analyse zu vereinfachen, werden zunächst Momentaufnahmen betrachtet: Welchen Namen trägt eine Person zu einem bestimmten Zeitpunkt? Wie verhält sich eine Person gegenüber anderen in Bezug auf Ehe und Familie? Bei der Rolle von Personen wird auch untersucht, wie Historie ins Spiel kommt.

Danach werden die Änderungen von Namen untersucht. Die Aussagen beschränken sich auf die zwei wichtigsten Teile des deutschen Namensrecht: die Namensänderung durch Eheschließung und Namensgebung bei Kindern. Andere Gründe für Namensänderungen können sein: Begründung einer Lebenspartnerschaft, Namensänderung bei Vertriebenen nach § 94 BVFG (Bundesvertriebenengesetz) etc. etc. All dies wird nicht betrachtet, weil es auch nicht Gegenstand der Musterlösung ist.

Schließlich wird der Metalevel (Knowledge Level) betrachtet.

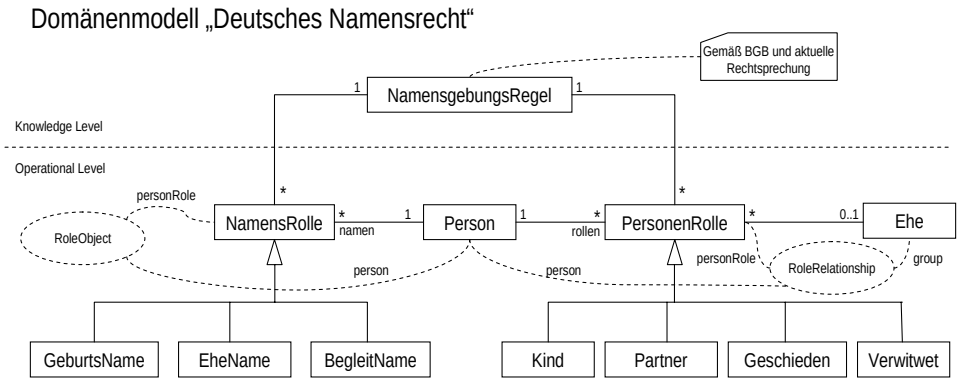

Abbildung 1: Musterlösung zum deutschen Namensrecht

2.1 Der Name einer Person – Momentaufnahme

Person und NamensRolle

Das Diagramm stellt eine Assoziation zwischen Person und NamensRolle dar. Gemeint ist offenbar, dass eine Person mehrere Namen haben kann im Laufe ihres Lebens. Zu einem Zeitpunkt (und jetzt interessiert ja zunächst die Mo-

mentaufnahme) kann es höchstens *ein* Name sein.[2] Dieser Sachverhalt geht aus dem Diagramm nicht hervor. Ich weiß nicht, ob die UML-Spezifikation eine Notation für den Unterschied zwischen Momentaufnahme und dem Einbezug der Historie vorsieht. Heide Balzert verwendet in ihren Büchern den Eigenschaftswert (*tagged value*) `{t = 0..1}`, um den Sachverhalt auszudrücken, dass zu einem Zeitpunkt höchstens ein Objekt assoziiert ist (siehe [1, S. 96]).

Wo aber ist der Name selbst? Ist er ein Attribut von NamensRolle? Dies ist zu vermuten, geht aber aus dem Diagramm nicht klar hervor. Gibt die Bezeichnung namen des Assoziationsendes einen Hinweis? In der UML 1.5 wird sie als Rollenname bezeichnet: „It indicates the role played by the class attached to the end of the path near the rolename.“ ([7, 3.43.2.6][3]. So gesehen muss man also wohl lesen: ‚Die Klasse NamensRolle spielt die Rolle des namens für die Person‘. Kann man diesen Satz einem Domänenexperten im Rahmen der Diskussion des Modells verständlich machen?

Gehen wir jedenfalls davon aus, dass der Name der Person ein Attribut der Klasse NamensRolle ist.

Im Diagramm gibt es drei Spezialisierungen der NamensRolle: GeburtsName, EheName und BegleitName. Wie ist dies zu interpretieren? Der Name, den eine Person zu einem Zeitpunkt hat, tritt in der Rolle des Geburtsnamens, Ehenamens oder Begleitnamens auf. Trifft das zu?

Eine Person kann (zu einem Zeitpunkt genau) einen Namen tragen, den Geburtsnamen oder den Ehenamen. Jedoch ist dies beim sogenannten Begleitnamen nicht so: Wenn Ehegatten einen Ehenamen wählen, dann kann der Ehepartner, dessen Geburtsname nicht Ehename wurde, dem Ehenamen seinen Geburtsnamen voranstellen oder anfügen (mit Bindestrich). Heiraten also `Gisela Bergmann` und `Johann Schneider`, dann können sie z.B. den Namen `Bergmann` zum Ehenamen bestimmen und `Johann` kann sich zusätzlich dazu entscheiden, den Namen `Schneider` als Begleitnamen voranzustellen oder anzufügen. Er trägt dann den Namen `Schneider-Bergmann` oder `Bergmann-Schneider`, aber nicht den Begleitnamen (`Schneider`) als Namen. Oder ist in dem Diagramm gemeint, dass die Zusammensetzung beider Namen als Begleitname zu bezeichnen ist? Woran erkannt man dann, ob der Geburtsname vorangestellt oder angefügt wurde?

Übrigens können auch bei Kindern zusammengesetzte Namen vorkommen:

> „Der Elternteil, dem die elterliche Sorge für ein unverheiratetes Kind allein oder gemeinsam mit dem anderen Elternteil zusteht, und sein Ehegatte, der nicht Elternteil des Kindes ist, können dem Kind, das sie in ihren gemeinsamen Haushalt aufgenommen

[2] Kurzzeitig hat eine Person auch keinen Namen: nämlich zwischen Geburt und Eintrag des Namens, das spielt aber hier keine Rolle.

[3] In der Version 2.0 der UML [6] taucht der Begriff Rollenname nicht mehr auf. Unklar weshalb!

> haben, durch Erklärung gegenüber dem Standesbeamten ihren Ehenamen erteilen. Sie können diesen Namen auch auch dem von dem Kind zur Zeit der Erklärung geführten Namen voranstellen oder anfügen...“ (§ 1618 BGB)

Der Name eines Kindes kann sich also aus dem Geburtsnamen und dem (neuen) Ehenamen z.B. der Mutter zusammensetzen. Könnte man diesen Sachverhalt dem Diagramm entnehmen?

Nun kann man das Diagramm vielleicht so interpretieren, dass all diese Informationen über die Namen in dem jedem Objekt der Klassen NamensRolle assoziierten Objekt der Klasse NamensgebungsRegel steckt. Dies wird später betrachtet.

Das Muster Role Object

Eine Information zum Thema Name einer Person bleibt noch zu hinterfragen: Es wird das Muster Role Object verwendet. Das Diagramm gibt an, dass in der Übertragung des Musters, die Klasse Person im Diagramm an die Stelle von person im Muster tritt und NamensRolle an die von personRole. Trifft das aber wirklich zu?

Fowler beschreibt Role Object so:

> „*How do you represent the many roles of an object?*
> Put common features on a host object with a separate object for each role. Clients ask the host object for the appropriate role to use a role's feature.“ ([3, S. 14])

Dieser Definition folgend wäre es aber der Name, der in unterschiedlichen Rollen auftritt, nicht die Person. Es gibt aber gar keine Klasse Name. Handelt es sich um eine Spezialform des Musters, bei dem das Objekt und seine Rollen in *derselben* Klasse modelliert sind? Wie ist aber dann die eingezeichnete Zuordnung des Musters zu verstehen?

(Man könnte noch weiter fragen: Wodurch unterscheiden sich die Klassen GeburtsName, EheName und BegleitName? Fowler empfiehlt das Muster Single Role Type, wenn sich die Rollen nicht sehr unterscheiden ([3, S. 17]). Die Art des Namens könnte dann ein Attribut dieser Klasse sein, die man dann auch einfach Name nennen könnte.)

2.2 Personen, Personenrollen und Ehe

Die Ehe – Momentaufnahme

Betrachten wir nun den rechten Teil des Diagramm auf dem Operational Level, die Ehe – zunächst wieder den Zustand zu einem Zeitpunkt.

Eine Person kann in Bezug auf eine Ehe verschiedene Rollen spielen, etwa Partner der Ehe sein oder Kind usw. Das Muster ist Role Relationship, erläutert etwa in [3, S. 17].

Hier ist die Multiplizität * in der Assoziation der Person zur PersonenRolle auch bei der Momentaufnahme angebracht, denn eine Person kann ja zur gleichen Zeit Partner in einer Ehe und Kind in einer anderen Ehe, der ihrer Eltern, sein.

In der zweiten Assoziation steht eine PersonenRolle in Beziehung zu höchstens einer Ehe. Um welchen Fall handelt es sich, wenn hier kein Objekt assoziiert ist? Möglicherweise um ein nicht ehelich geborenes Kind? Dann bräuchte man aber für Belange der Namensgebung (mindestens) die Beziehung dieses Kinds zu seiner Mutter. Oder sind nicht verheiratete Partner gemeint? Woran erkennt man dann aber ihre Partnerschaft?

Oder ist der Fall der Scheidung oder des Tods eines Partners gemeint? Bei dieser Interpretation stellt das Diagramm in seinem rechten Teil *nur* eine Momentaufnahme dar – im Unterschied zum linken Teil, bei dem Historie mit gemeint ist. Woran kann der Leser des Diagramms diesen Unterschied erkennen?

Darüberhinaus bleiben Fragen: Kann eine Ehe mehr als zwei Partner haben? Kann eine Person zum gleichen Zeitpunkt mehrere Ehen führen? Dies sind sicherlich Angaben, die man durch Ausdrücke der *Object Constraint Language* darstellen kann. So kann man festlegen, dass eine Ehe genau zwei Partner hat:

```
context e:Ehe inv:
  e.personenRolle ->
    select( pr | pr.OclIsKindOf(Partner) ) -> size() = 2
```

Aber sind OCL-Ausdrücke ein geeignetes Mittel der Kommunikation mit Domänenexperten oder Kunden?[4]

Die Ehe – mit Historie

Soweit die Fragen zur Momentaufnahme. Betrachten wir noch kurz die Historie der Gruppe Ehe: Wird historische Information berücksichtigt, dann sollte das Diagramm ausdrücken, dass eine Person auch mehrfach verheiratet sein kann. Das könnte man so machen, dass ein Objekt der Klasse Person die PersonenRolle Partner nicht nur in Bezug auf ein, sondern auf mehrere Objekte der Klasse Ehe spielt. Informationen über Zeiträume des Bestehens der Ehe könnte man in den Klassen Ehe oder auch PersonenRolle unterbringen.

Dann ergibt sich aber, ob eine Person geschieden ist, daraus, dass die Rolle Partner eine gewisse Zeit bestanden hat, jetzt aber nicht mehr. Wenn man das Diagramm so auffasst, dann fragt sich, wieso es die Klassen Geschieden und Verwitwet gibt. Sie machen nämlich eigentlich nur Sinn, wenn man die Momentaufnahme einer Konstellation betrachtet.

[4]Der Fairness halber sei hinzugefügt, dass dies wahrscheinlich bei jeder formalen Beschreibung so ist. Man braucht also meist die formale Beschreibung mitsamt einem erläuternden Text in natürlicher Sprache.

Auch aus dieser Sicht stellt sich also die Frage: Ist der rechte Teil des Diagramms für eine Momentaufnahme gedacht oder für die Historie der Personen und ihrer Ehen?

2.3 Namensänderung durch Eheschließung

Bei der Namensänderung durch Eheschließung kommt natürlich zwangsläufig die Historie ins Spiel, weil der Name einer Person nach der Ehe davon abhängt, wie die Lage vor der Ehe war – und was die Eheschließenden entschieden haben.

Die erste Frage ist, ob die Ehepartner einen gemeinsamen Ehenamen bestimmen. Wenn sie das nicht tun, besteht getrennte Namensführung und sie behalten ihre bisherigen Namen. D.h. ein Ehename kommt durch die gemeinsame Entscheidung der Eheschließenden zustande. Kann man diesen Sachverhalt dem Diagramm entnehmen?[5]

Wenn sie einen Ehenamen bestimmen, dann kann der Partner, dessen Name nicht der Ehename wurde, seinen Geburtsnamen dem Ehenamen voranstellen oder anfügen. Im Diagramm erkennbar?

Bisher war es so, dass nur der Geburtsname einer Person Ehename werden konnte, nicht der Name aus einer vorherigen Ehe.[6] Nach einem Urteil des Bundesverfassungsgerichts muss der Gesetzgeber bis zum 31. März 2005 das Gesetz ändern. Dies erweitert die Möglichkeiten bei der Wahl des Namens in der Ehe beträchtlich. Hätte es Änderungen in unserem Diagramm zur Folge? Man sollte dies erwarten, es scheint aber nicht der Fall zu sein.

2.4 Namensgebung bei Kindern

Bei der Namensgebung von Kindern aus einer Ehe ist entscheidend, ob die Eltern einen gemeinsamen Ehenamen haben. Ist dies der Fall, dann bekommt das Kind den Ehenamen als Geburtsnamen. Ist das nicht der Fall, dann können die Eltern einen ihrer Namen als Geburtsnamen des ersten Kindes bestimmen. Alle später geborenen Geschwister dieses ersten Kindes bekommen dann auch diesen Namen. Die Eltern entscheiden sich also beim ersten Kind für den Namen *aller* ihrer gemeinsamen Kinder. Sind diese Sachverhalte im Diagramm erkennbar?

Bei nicht ehelich geborenen Kindern mit gemeinsamem Sorgerecht der Eltern wird wie eben beschrieben verfahren. Hat nur ein Elternteil das Sorgerecht, bekommt das Kind den Namen, den dieser Elternteil zum Zeitpunkt der Geburt hatte. Aber: Willigt der nicht sorgeberechtigte Elternteil ein,

[5] Man könnte übrigens meinen, dass im Fall der Bestimmung eines gemeinsamen Ehenamens beide Partner der Ehe *denselben* Namen tragen. Dies muss aber nicht wirklich der Fall sein. Er könnte `Freiherr zu Wittgenstein` und sie `Freifrau zu Wittgenstein` heißen.

[6] Wahrscheinlich hat der Adel seinen Einfluss bei dieser Regelung geltend gemacht, weil er verhindern wollte, das eingeheiratete bürgerliche Partner den edlen Namen in eine rein bürgerliche Verbindung weitertragen können.

dann kann der sorgeberechtigte Elternteil auch dessen Namen zum Geburtsnamen des Kindes bestimmen (§ 1617a BGB). Und was der Bestimmungen mehr sind... Könnte man Fragen danach, welche Namen in einer bestimmten Situation für ein Kind nach dem deutschen Namensrecht erlaubt sind, mit Hilfe des Diagramms beantworten?

2.5 Der Knowledge Level

Vielleicht sind all diese Fragen über das Namensrecht gar nicht zu stellen, weil das Modell auf dem Operational Level gar nicht beabsichtigt, Aussagen über das Zustandekommen der Namen zu machen. Der Kommentar zur Klasse NamensgebungsRegel spricht ja all die gesetzlichen Regelungen an, auf die sich ein Teil der bisherigen Fragen bezogen haben. Betrachten wir also den Metalevel:

Zu jeder NamensRolle und PersonenRolle gibt es genau eine Regel. Zum Beispiel hat jedes Objekt der Klasse Geburtsname eine zugeordnetes Objekt der Klasse NamensgebungsRegel, das wiederum mit eventuell mehreren Objekten der Klasse PersonenRolle verlinkt sein kann. Wie kann man dieses Modell interpretieren?

Im Diagramm sind keine Bezeichnungen für die genannten Assoziationen angegeben. Die linke Assoziation zwischen NamensgebungsRegel und NamensRolle könnte bedeuten „bestimmt“. Somit: ‚Die jeweils zugeordnete Regel bestimmt diese Namensrolle, damit den Namen‘. Die rechte Assoziation zwischen NamensgebungsRegel und PersonenRolle könnte bedeuten „nimmt Bezug auf“. Somit: ‚In der Bestimmung des Namens durch diese Regel wird auf die und die Personenrolle Bezug genommen‘.

Betrachten wir als Beispiel (Abb. 2) die Familie `Dominik`, bestehend aus dem Vater `Andreas Dominik`, der Mutter `Elisabeth Dominik` und der Tochter `Johanna`. Die Eheleute haben sich für einen gemeinsamen Ehenamen entschieden, den Geburtsnamen des Mannes, `Dominik`. `Elisabeth` hatte als Geburtsnamen den Namen `Renz`. `Johanna` erhält als Geburtsnamen den Ehenamen der Eltern, also `Dominik`.

Auf dem Operational Level haben wir die Personen p1, p2 und p3, also `Andreas`, `Elisabeth` und `Johanna`. Sie spielen in Bezug auf die Ehe e die eingezeichneten Rollen als Partner bzw. Kind. An Namen kommen ins Spiel: die Geburtsnamen der beiden Ehepartner, der gemeinsame Ehename und der Geburtsname der Tochter.

Wie sieht es nun auf dem Knowledge Level aus? Folgt man der Auslegung des Diagramms, wie oben beschrieben, ergibt sich:

Die NamensgebungsRegel r1 bezieht sich auf das Thema Ehename und beinhaltet etwa die Aussage ‚Wenn die Ehepartner einen gemeinsamen Ehenamen bestimmen, dann ist dies einer ihrer Geburtsnamen‘. Die Verbindung von r1 zum Objekt e:Ehename bedeutet also, dass dieser Name durch die Regel bestimmt wird. Die Regel r1 hat eine Verbindung zu den PersonenRollen v und m. Dahinter steckt die Überlegung, dass in der Regel zur Bestimmung

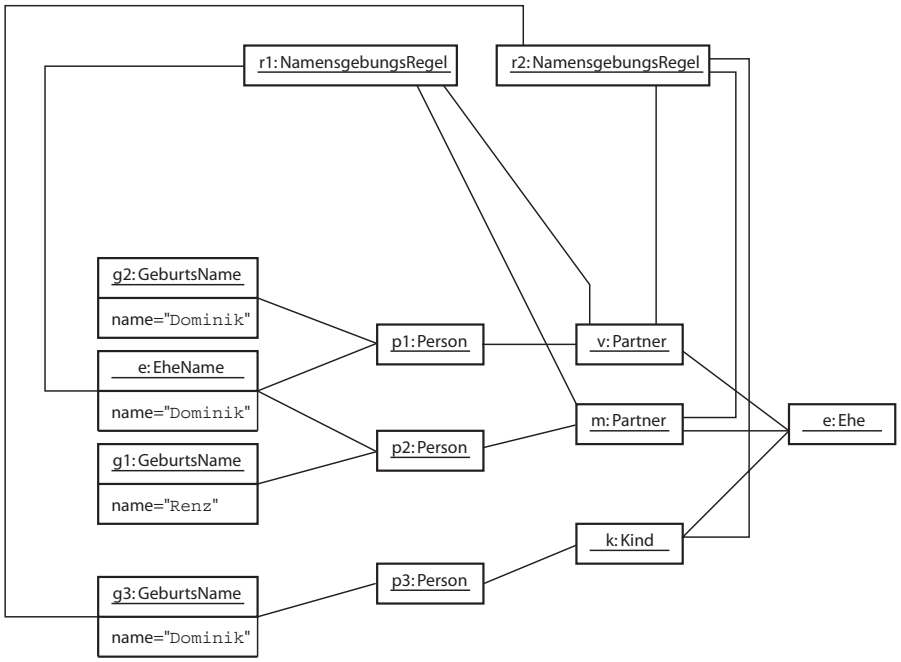

Abbildung 2: Objektdiagramm: Familie `Dominik`

des Ehenamens auf die Rolle der Beteiligten als Partner in einer Ehe Bezug genommen wird.

In Abb. 2 kommt eine weitere Regel vor: r2. Sie könnte etwa beinhalten: ‚Kinder von Ehepartner, die einen Ehenamen tragen, erhalten diesen als Geburtsnamen'. Analog zur Situation beim Ehenamen ist im Beispiel eine Verbindung zwischen dem Objekt g3 und der Regel r2 eingezeichnet, weil die Regel den Geburtsnamen des Kindes bestimmt. Ferner gibt es Verbindungen zu den Personenrollen von Vater, Mutter und Kind, weil die Regel auf diese Personenrollen Bezug nimmt.

Nun ergibt sich aber ein Widerspruch zum Klassendiagramm, denn das Objektdiagramm in Abb. 2 erfüllt die Vorgabe der Multiplizitäten im Klassendiagramm in Abb. 1 nicht. Wie kann man das Diagramm aber sonst interpretieren? Wieviele Interpretationen erlaubt es? Die Verwendung des Modells für die Softwareentwicklung erfordert doch eindeutige, klare Interpretationen – ist das mit UML-Diagrammen für so komplexe Problembereiche überhaupt möglich?

An dem Beispiel ergibt sich außerdem eine grundsätzliche Frage: Wie ist die Klasse NamensgebungsRegel aufgebaut?

Die Frage zielt auf die Eigenart von Metalevel-Konstruktionen, dass die *Objekte* der Metaebene die *Klassen* der Basisebene beschreiben. In der UML etwa sind Modellelemente Exemplare (*instances*) von Klassen des Metamo-

dells. In unserem Fall sind es doch die *tatsächlichen* Regeln des Namensrechts (nicht ihr struktureller Aufbau), die den Operational Level beherrschen.

Man kann natürlich ein Klassenmodell bauen, mit dem man Ausdrücke, Operatoren, Vergleiche u.ä. darstellen kann. Das würde aber nicht wirklich helfen. Um das Namensrecht zu beschreiben, bräuchte man doch Objekte solcher Klassen; Objekte, die Ausdrücke der Aussagenlogik wie etwa ‚Eltern haben Ehenamen impliziert Geburtsname des Kindes ist identisch mit Ehename' darstellen.

Ist dafür die Klasse NamensgebungsRegel gedacht? Wie soll sie dann solche Sachverhalte ausdrücken? Und ferner: In welchem Verhältnis steht diese Klasse zur OCL? Das Prinzip der OCL ist es ja gerade, Aussagen über Objekte zu machen: Invarianten, Beziehungen zu anderen Objekten usw. Wie grenzt man die OCL vom Knowledge Level ab?

Festzuhalten bleibt, dass die Regeln des Namensrechts in dem Modell nicht vorkommen. Infolgedessen kann man auch unter Zuhilfenahme des Knowledge Levels keine Aussagen über die Namensgebung bestätigen oder falsifizieren. Wie soll dann aber das Modell in einer Diskussion mit Domänenexperten hilfreich sein?

Offenbar ist es nicht die Intention des Modells, solche Aussagen zu machen. Vielleicht ist eine Abstraktionsebene gewählt, bei der solche Aussagen „wegabstrahiert" sind? Handelt es sich aber dann um eine Abstraktion? Abstraktionen sollen doch das Wesentliche, das Wichtige zeigen – was aber, wenn sie die Aussagekraft verkleinern: eine *leere* Abstraktion? Wozu dient aber dann das Modell?

3 Schlussfolgerungen

3.1 Nutzen der Musterlösung für die Softwareentwicklung

Will man eine Software für das Personenstandswesen konstruieren, dann braucht man eine Beschreibung des deutschen Namenrechts – und man braucht ein Modell in der Software. Fassen wir als Ergebnis zusammen, was die Musterlösung dafür leistet.

Sie reißt die wesentlichen Punkte an: den Stand der Personen, gerade zueinander; die Arten von Namen und ihre Bestandteile; die gesetzlichen Bestimmungen, die die Namensgebung bestimmen; die Historie im Personenstand und in den Namen einer Person.

Andererseits bleibt sie vage, ist mehrdeutig, lässt sehr, sehr viel Interpretationsspielraum. Sie beantwortet viele der Fragen, die bei der Entwicklung einer Software anfallen würden, nicht.

Was die Beschreibung des Namensrechts angeht: Die Analyse sollte nicht nur den Inhalt der gesetzlichen Bestimmung schematisieren, sondern auch auf Lücken (die ja Fallstricke für die Software sein können) hinweisen. Das Beispiel zeigt meines Erachtens, dass der Ansatz der objektorientierten Analyse und die Notation in UML für *diese* Aufgabe nicht wirklich geeignet sind.

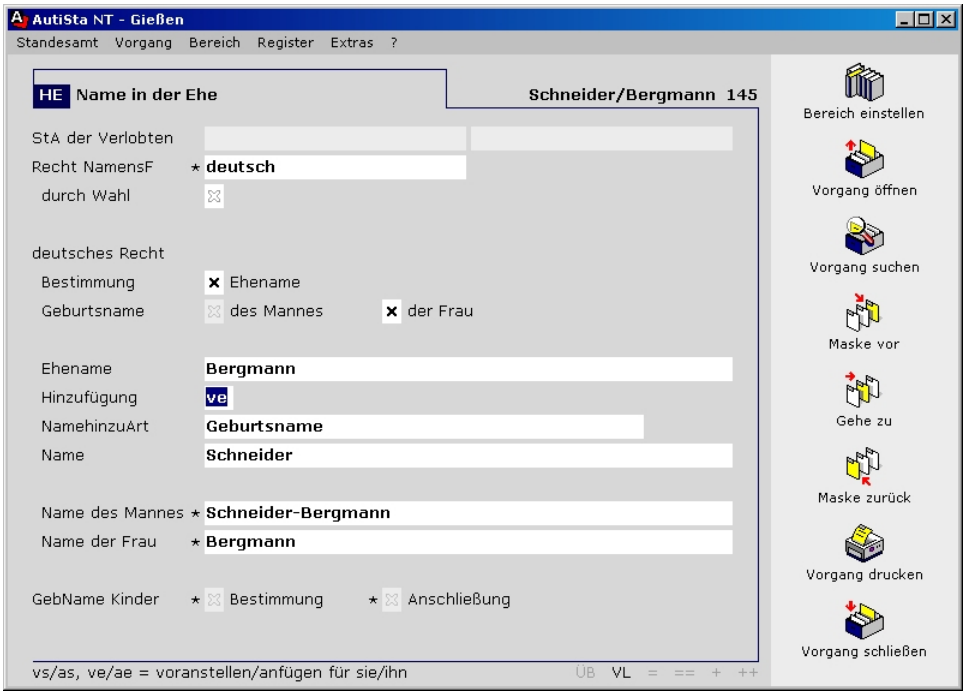

Abbildung 3: Eingabemaske für Ehenamen nach deutschem Recht in der Standardsoftware AutiSta (Automation im Standesamt) des Verlags für Standesamtswesen, Frankfurt am Main, Berlin

Was das analoge Modell angeht: Bezieht man es auf eine Software, die die Namensart, Namensbestandteile und Bestimmungsgründe dafür angibt, ist es unvollständig; wiewohl ein Objektmodell natürlich denkbar ist. Bezieht man den Metalevel, d.h. die Modellierung der Regeln selbst mit ein, wird das Modell sehr komplex – und bietet in seiner derzeitigen Form keine geeignete Möglichkeit die Tatsache auszudrücken, dass es die *Objekte* des Metalevels sind, die die Struktur und Beziehungen der *Klassen* des Basislevels bestimmen.

Würden zwei Softwareentwickler (unabhängig voneinander) auf Basis der Musterlösung eine Software für das Personenstandswesen konstruieren, kämen sie wahrscheinlich zu ganz unterschiedlichen Lösungen, und zwar zu solchen, die nicht mehr viel Ähnlichkeit mit dem Modell der Musterlösung hätten. Oder sie würden eine direkte Abbildung der Analyseklassen in Designklassen vornehmen. Dann hätten sie wohl ähnliche Lösungen, aber wofür würde die darauf basierende Software taugen?

3.2 „Und wo bleibt das Positive, Herr Kästner?“

Eine Beschreibung des deutschen Namensrechts kann man dadurch hinbekommen, dass man in Tabellenform alle Kombinationsmöglichkeiten aufschreibt, in denen die beteiligten Personen (Ehepartner, Kinder und Eltern) ankommen, welche Bestimmungsmöglichkeiten sie haben, und welches Ergebnis dann in der Namensgebung herauskommt. Ein solches Schema kann für die Diskussion mit Domänenexperten mit konkreten Fallbeispielen versehen werden, die die Überprüfung des Schemas erleichtern. (Später können diese Fallbeispiele auch für die Überprüfung der zu konstruierenden Software verwendet werden.) Die Schematisierung kann auch helfen, die Fälle zu erkennen, in denen die Rechtslage unklar oder umstritten ist.

Man könnte versucht sein, eine Notation zu finden, in der man diese Sachverhalte als logische Ausdrücke darstellen kann, so dass man sogar aus bekannten Sachverhalten neue ableiten kann. (Meine Erfahrung ist, dass dieser Ansatz zwar möglich ist, aber sich zur Diskussion mit Domänenexperten nicht eignet.)

Aus einer Beschreibung des Namensrechts kann man dann Modelle je nach Anforderung der zu konstruierenden Software bilden. Für eine Software im Bereich Personenstandswesen wird ein Modell benötigt, das zumindest das Zustandekommen des Namens exakt *dokumentiert*. Denn der Standesbeamte muss eben dies beurkunden.

Wie ein solches Modell in Aktion aussehen kann, zeigt die Bildschirmmaske in Abb. 3. Aus den Eingabefeldern wird ersichtlich, welche zugrunde liegenden Objekte (Datenfelder) benötigt werden. Ein Modell auf diesem Level kann nicht den Anspruch erheben, die Regeln des Namensrechts abzubilden – denn es kann nicht prüfen, ob die Konstellation tatsächlich erlaubt ist.

Will man diese Art Prüfung auch noch in einer Software abbilden, benötigt man einen Beschreibungsbereich (*description domain*, siehe [4, Chap. 8.2]), der die Regeln des Namensrechts in einer operationalisierten, durch Software verarbeitbaren Form enthält – ein äußerst anspruchsvolles Unterfangen. In der Software, die in Abb. 3 gezeigt wird, ist ein Beschreibungsbereich enthalten mit logischen Ausdrücken, die zur Laufzeit durch einen Interpreter ausgewertet werden. Allerdings handelt es sich nur um die einfachsten logischen Regeln, nur um einen Bruchteil dessen, was für das Namensrecht benötigt wird. Die Software überlässt die Kenntnis dieser Regeln den Anwendern im Standesamt; der eingebaute Regelinterpreter soll nur helfen, grobe Fehler in der Bearbeitung von Personenstandsfällen zu vermeiden.

3.3 Folgerungen für die Lehre

- Bei den Studierenden muss das Bewusstsein geschärft werden für die Tatsache, dass nur dann qualitativ hochwertige Software konstruiert werden kann, wenn der *Problembereich* verstanden wird. Für dieses

Verständnis gibt es keinen Königsweg; wie man es erlangt, hängt vom jeweiligen Fachgebiet ab. Freilich kann man in einer Veranstaltung zum Thema Softwaretechnik kein spezielles Fachgebiet darstellen, so dass dieses Bewusstsein nur dadurch entsteht, dass klar unterschieden wird, zwischen der Sache, den Problemen der wirklichen Welt und den Konstrukten der Softwaretechnik.

- Objektorientierung ist *eine* Methode, die ihren Wert insbesondere in der Konstruktion von Codestrukturen hat. Für die Beschreibung des Problembereichs ist sie nur in bestimmten Fällen nützlich. Die UML kann für die Visualisierung von Sachverhalten verwendet werden, sie allein reicht für die Kommunikation mit Domänenexperten nicht aus. In der Lehre sollte vermittelt werden, dass verschiedene Probleme verschiedene Techniken der Beschreibung brauchen, man muss als Softwareentwickler jeweils eine *geeignete* Methode finden.

Literaturverzeichnis

[1] Balzert, Heide *Lehrbuch der Objektmodellierung: Analyse und Entwurf* Spektrum Akademischer Verlag 1999.

[2] Fowler, Martin *Analysis Patterns: Reusable Object Models* Addison-Wesley 1999.

[3] Fowler, Martin *Dealing with Roles* <www.martinfowler.com> 1997.

[4] Jackson, Michael *Problem Frames: Analysing and structuring software development problems* Addison-Wesley 2001.

[5] Meyer, Bertrand *Object-Oriented Software Construction, second edition* Prentice Hall 1997.

[6] OMG *Unified Modeling Language: Superstructure version 2.0 Final Adopted Specification 03-08-02* <www.omg.org> 2003.

[7] OMG *Unified Modeling Language Specification March 2003 Version 1.5* <www.omg.org> 2003.

[8] Rational Software Corp. *Rational Unified Process* CD von Rational Software Corp. 2000.