Dr. Christoph Lübbert
Viktoriastraße 36
D-64293 Darmstadt

Tel: 06151 422298, T-Mob: 0171 2045811,
christoph.luebbert@t-online.de ,
www.cl-diesunddas.de

Darmstadt, 15.07.2014

Vermittlung zwischen Onto-Sicht und FBA-Sicht im Konzept „Ontology“

Inhalt

Aufgrund der Diskussionen, die auf meinen Vortrag [2] ***„Verfeinerung der Ontology-Sprache durch die FBA-Sprache“*** am 23.6.14 folgten, halte ich einige kleine Erläuterungen und Ergänzungen für angebracht.

1 Wiederholung aus dem Vortrag [2] von CL

Aus dem Vortrag [2] am 23.6.14 sei für das Konzept „Ontology“ Folgendes kurz wiederholt: Das mir bekannte konventionelle Konzept „Ontology“ (vgl. zum Beispiel *Maedche* et.al. [3]) geht nun einmal von folgenden Komponenten aus, die ich an sich nicht kritisieren sondern nur etwas aus FBA-Sicht „verfeinern“ / „präzisieren“ wollte, – nämlich:

• Im **O-Schema:** Eine Menge $\mathbb{C}$ von „*Klassen*“, „*Klassentaxonomie*“ $<_C$ auf $\mathbb{C}$, eine Menge $\mathbb{A}$ von „*Attributen*“, *Operator att* (der jeder Klasse c eine Attributemenge att(c) zuweist), eine Menge **RELT** von (binären) „*Relationship-Types*“ (zwischen 2 Klassen), *Operator relt* (der jedem Relationship-Type r ein gewisses Paar relt(r)=(c, d) von Klassen zuweist), sowie das zentrale Prinzip der ***Vererbung*** von Attributen und Relationship-Types entlang der Klassen-taxonomie $<_C$.

• In der **Instanziierungsebene:** „*Instanzen*“, *Operator inst* (der jeder Klasse c eine Menge inst(c) von Instanzen zuweist), „*Attributwerte*“ (mit einem Attribut a∈ att(c) wird einer Instanz g∈ inst(c) eindeutig ein Attributwert a(g) zugewiesen).

An diese ***„Ontology-Sprache“*** hatte ich mich im Vortrag erst einmal gehalten. Ich hatte nur gezeigt, dass man unter Berücksichtigung der Tatsache, dass die Klassentaxonomie $<_C$ eine ***Ordnung*** (irreflexive, transitive binäre Beziehung zwischen Klassen) sein soll, man die Operatoren **att** und **inst** als ***Ordnungshomomorphismen* präzisieren** kann, so dass man in natürlicher Weise auf die ***„FBA-Sprache“*** kommt, wodurch das taxonomisch geordnete Klassensystem $\mathfrak{C}$ und das zugehörige Attributemengensystem $[\mathfrak{A}] := \{att(c) | c \in \mathfrak{C}\}$ sich als zwei zueinander duale ***vollständige Verbände*** erweisen. Ferner hatte ich gezeigt, dass die Gesamtheit **RELT** der im O-Schema gebrauchten Relationschip-Types ebenfalls in natürlicher Weise geordnet ist; diese Ordnung hatte ich die ***Relationship-Taxonomie*** genannt. Weiter hatte ich gezeigt, wie man eine durch $relt(r)=(c,d)$ $[r \in \mathbf{RELT}; c,d \in \mathfrak{C}]$ gegebene relationale Teilkonfiguration **c r d** des O-Schemas durch einen formalen Kontext ($\underline{c}$, $\underline{d}$, $\underline{r}$) darstellt und sie durch die *formalen Begriffe* des zum Kontext gehörigen *formalen Begriffsverbandes* $\underline{\mathbf{B}}(\underline{r}, \leq_r)$ „verfeinern“ kann. Schließlich hatte ich gezeigt, dass die Zuweisung der *Attribut**werte*** zu Instanzen in der FBA-Sprache durch einen ***mehrwertigen formalen Kontext*** dargestellt werden kann.

Hier noch einmal kurz die Komponenten dieses konventionellen Konzepts „Ontology“:

(a) $(\mathfrak{C}, <_C)$ ist das **geordnete** System der **Klassen** (das „Rückgrad“ eines O-Schemas) mit $<_C$ als **Klassentaxonomie** ($<_C$ ist eine *irreflexive* und *transitive* binäre Beziehung, d.h. eine *strikte Ordnung* auf $\mathfrak{C}$; die zugehörige nicht-strikte Form $\leq_C$ ist durch $x \leq_C y :\Leftrightarrow (x <_C y \text{ oder } x=y)$ definiert);

(b) $\mathfrak{A}$ die Menge der **Attribute**; att: $\mathfrak{C} \rightarrow \text{Pot}\mathfrak{A}$ der Operator, welcher jeder Klasse $c \in \mathfrak{C}$ ihre Attributemenge $att(c) \subseteq \mathfrak{A}$ zuordnet;

(c) inst: $\mathfrak{C} \rightarrow \text{Pot}\mathbf{G}$ der Operator, welcher jeder Klasse $c \in \mathfrak{C}$ die Menge $inst(c) \subseteq \mathbf{G}$ der zu c ermittelbaren **Instanzen** zuordnet. Mit **G** sei die Gesamtheit der in der Ontology ermittelbaren Instanzen bezeichnet.

(d) Ähnlich wie in der OOP hat man im O-Schema als fundamentales Prinzip die sog. **Vererbung der Attribute *von oben nach unten*** entlang der Klassentaxonomie $<_C$: Ist $c <_C d$, so werden an die Unterklasse c alle Attribute der Oberklasse d **vererbt**; die Unterklasse c darf darüber hinaus noch weitere Attribute haben, welche die Oberklasse d nicht hat. Je *spezieller* die Klasse, desto *mehr* Attribute hat sie i.allg. In Formeln heißt das:

(1) $att(c) = att_{vererbt}(c) \cup att_{neu}(c)$ für jede Klasse $c \in \mathfrak{C}$; (evtl. darf $att_{neu}(c) = \emptyset$ sein),

(2) $att(d) \subseteq att(c)$ für jedes Paar c, $d \in \mathfrak{C}$ mit $\mathbf{c <_C d}$.

Hierüber gab es keinen Widerspruch von Seiten der Informatiker.

Analoges bei den Instanzen: Ist $c <_C d$, so heißt das: die Klasse c ist ***spezieller*** als die Klasse d (oder, was dasselbe bedeutet: die Klasse d ist ***allgemeiner*** als die Klasse c). Je spezieller die Klasse, desto weniger ihre Instanzen i.allg. Bei $c <_C d$ können also der Unterklasse c jedenfalls *nicht mehr* Instanzen zugeordnet werden als der Oberklasse d, denn es findet sozusagen eine Vererbung der Instanzen ***von unten nach oben*** entlang der Klassentaxonomie $<_C$ statt. In Formeln heißt das:

(3) $inst(c) \subseteq inst(d)$ für jedes Paar c, $d \in \mathfrak{C}$ mit $\mathbf{c <_C d}$.

Auch hierüber gab es (nach meiner Erinnerung) keinen Widerspruch von Seiten der Informatiker.

2 Widerspruch von Seiten der Informatiker

Im meinem Vortrag gab es **Widerspruch** von Seiten einiger Informatiker gegen folgende über (1), (2), (3) hinausgehende schärfere Forderung von CL an eine „Ontology“:

(4) att(c) = att(d) ⇔ c = d für alle Klassen c, d ∈ ℂ des O-Schemas,
(5) inst(c) = inst(d) ⇔ c = d für alle Klassen c, d ∈ ℂ des O-Schemas.

In Worten: Zwei Klassen sind (4) genau dann gleich, wenn sie in ihren Attributen übereinstimmen; und (5) genau dann gleich, wenn sie in ihren Instanzen übereinstimmen.

Bisher habe ich als einzige Begründung gegen die Präzisierungen (4), (5) gehört, dass sie „die Design-Arbeit der Ontology-Ersteller zu sehr einschränke".

Meine Entgegnung: Das genügt mir als Mathematiker aber ***bei weitem nicht!*** Ich vermisse eine logisch-innersystemische Begründung gegen (4), (5). Eine solche wurde mir bislang von den Informatikern unseres Kreises nicht gegeben. Das aber deutet mir darauf hin, dass das bisherige (wie oben geschilderte) konventionelle Konzept „Ontology" für eine konsistente Programmierung ***noch nicht vollständig durchdacht*** worden ist. Nach meiner Vermutng stützt sich der Widerspruch der Informatiker gegen meine Präzisierung (4), (5) auf ein unausgereiftes OOP-Konzept der 1960-er Jahre oder andere Datenkonzepte, die für spezielle Zwecke zeitlich ***vor*** dem Konzept „Ontology" entstanden sind.

Zeit-Problem: Ferner wurde eingewendet, dass meine Präzisierungen „statisch" seien und nicht berücksichtigen, dass sich das Konzept „Ontology" auch an mögliche ***Erweiterungen*** der Implementierung der darzustellenden Wissengebiete im Verlauf der Zeit anzupassen habe. Das ist richtig.
Das Zeit-Problem kann aber gelöst werden, wenn für alle Komponenten einer „Ontology" ein diskreter „***Zeitparameter***" – und damit eine **Versionsnummer** „t" – eingeführt wird und untersucht wird, in welcher „Ontology"-Variante die „Deltas" zwischen Ontology(t) und Ontology(t+1) ***mathematisch*** wohl-definiert und am einfachsten handhabbar werden (und eben ***nicht mehr*** nur von irgendwelchen bisher üblichen Datenmodellierungskonventionen – etwa ERM o.ä. – abhängig gemacht werden). Hierauf gehe ich in dieser Note noch nicht ein. In der O-Definition [4] / Kap.3.3.7 aber hatte ich das Problem angesprochen und zu einem gewissen Grad gelöst.

3 Ein Kompromissvorschlag

Der hier vorgeschlagene Kompromiss bzgl. des erhobenen Widerspruchs der Informatiker unseres ONTO-Kreises ist vorerst „statisch", d.h. er berücksichtigt noch nicht das Zeit-Problem (also die Versionsführung einer „Ontology").

Ich halte zunächst einmal an den nicht widersprochenen Formeln (1), (2), (3) und ebenso an der Präzisierung (5) für die Instanziierung fest. Die Präzisierung (4), zu der es den ausgeprägtesten Widerspruch gab, will ich nun modifizieren.

3.1 Natürliche Erweiterung des Attribute-Bereichs

Thomas Zeh hat vorgeschlagen, jede Klasse c nicht nur durch ihre bisherige Attributemenge att(c) sondern auch durch die eventuelle Beteiligung an einem Relationship-Type (in Form einer Teilkonfiguration **c r d** oder auch **d s c** mit c, d ∈ ℂ , r, s∈ **RELT**) zu charakterisieren.

Dazu wollen wir erst einmal präzisieren, wie der Operator relt: **RELT** → ℂ × ℂ zu bilden ist. Zu jedem binären Relationship-Type r∈**RELT** gehört die ***Aussageform*** ϒ (auf Instanzenebene) und die Relation (im mathematischen Sinne) $\underline{r}$:= {(g,h)∈**G**×**G**| g ϒ h}, wobei **G** wieder die Gesamtheit aller in der Ontology ermittelbaren Instanzen sei: **G** = $\mathbf{U}_{c\in\mathbb{C}}$ inst(c). $\underline{r}$ ist eine gewisse *Teilmenge* von **G**×**G,** d.h. $\underline{r}\subseteq$**G**×**G.** Statt „inst(c)" schreiben wir kürzer „$\underline{c}$".

Definition 1: Zu Vereinfachung der Formulierung führe ich auf dem Klassenprodukt ℂ × ℂ mit Hilfe der Klassentaxonomie $\leq_C$ bzw. $<_C$ folgende nicht-strikte Ordnung „$\leq^2_C$" bzw. die strikte Ordnung „$<^2_C$" ein:

$(c, d) \leq^2_C (c', d') \;:\Leftrightarrow\; (c \leq_C c' \text{ und } d \leq_C d')$,
$(c, d) <^2_C (c', d') \;:\Leftrightarrow\; ((c, d) \leq^2_C (c', d') \text{ und } (c <_C c' \text{ oder } d <_C d'))$
für alle $c, c', d, d' \in \mathbb{C}$.

Definition 2: Gegeben ein Relationship-Type $r \in$ **RELT**. Jedes Klassenpaar (c, d) mit der Eigenschaft $\underline{r} \subseteq \underline{c} \times \underline{d}$ nennen wir einen „**Rahmen**" für r; wir sagen auch *„Der Relationship-Type r ist im Rahmen (c, d) aufgehängt"*. Dann ist, wenn wir die TopClass durch $\underline{\text{TopClass}}$:= **G** definieren, (TopClass, TopClass) jedenfalls ein Rahmen für *jeden* Reltionship-Type r. Das trifft die Sache aber noch nicht: Zu gegebenem $r \in$ **RELT** gibt es nun einen **kleinsten Rahmen** $\mathbf{(c_r, d_r)} \in \mathbb{C} \times \mathbb{C}$, in dem r *aufgehängt* ist, so dass für jedes Klassenpaar (c', d') aus (c', d') $<^2_C$ (c_r, d_r) folgt: $r \not\subseteq \underline{c'} \times \underline{d'}$. Nun definieren wir den Operator relt durch **relt(r) := (c_r, d_r)**. relt(r) bestimmt also den ***kleinsten Rahmen***, in dem der Relationship-Types r aufgehängt ist. Damit ist der Operator relt: **RELT** $\rightarrow \mathbb{C} \times \mathbb{C}$ im O-Schema wohl-definiert.

Definition 3: Sodann definieren wir gewisse **Zusatzattribute**, mit denen wir die ursprüngliche Attributemenge $\mathbb{A}$ zu einer Attributemenge $\mathbb{A}^+$ bzw. den ursprünglichen Operator att zu einem Operator $att^+: \mathbb{C} \rightarrow [\mathbb{A}^+]$ erweitern:

- Ist c eine Klasse, zu der es eine Klasse d und einen ***maximalen*** Relationship-Type $r \in$ **RELT$_{max}$** gibt mit relt(r)=(c, d), so geben wir der Klasse c das Zusatzattribut
  a_{1r} := **.**Pos1_in_r:…. . d.h.: $a_{1r} \in att^+(c) \subseteq \mathbb{A}^+$.
- Ist aber c eine Klasse, zu der es eine Klasse d und einen ***maximalen*** Relationship-Type $s \in$ **RELT$_{max}$** gibt mit relt(s)=(d, c), so geben wir der Klasse c das Zusatzattribut
  a_{2s} := **.**Pos2_in_s: …. . d.h.: $a_{2s} \in att^+(c) \subseteq \mathbb{A}^+$.

Diese Zusatzattribute werden – genauso wie die ursprünglichen Attribute – entlang der Klassentaxonomie $<_C$ von oben nach unten **vererbt**.

Anmerkung: Zu demselben Klassenpaar (c, d) darf es u.U. natürlich auch *mehrere* sowohl vererbte als auch maximale Relationship-Types r, s, … mit relt(r) = relt(s) =… = (c, d) geben.

Was sollen nun (bei Instanziierung) die ***Werte*** dieser Zusatzattribute sein?

Definition 4: Sei g eine Instanz der Klasse $c \in \mathbb{C}$, $g \in \underline{c}$ =inst(c). Hat nun c das Zusatzattribut a_{1r}, und gibt es eine Instanz $h \in$ **G** mit g ϒ h, d.h. $(g,h) \in \underline{r}$, so sei $a_{1r}(g)$:=1 gesetzt, andernfalls sei $a_{1r}(g)$:=0 gesetzt. Hat aber c das Zusatzattribut a_{2s}, und gibt es eine Instanz $h \in$ **G** mit h ϡ g, d.h. $(h, g) \in \underline{s}$, so sei $a_{2s}(g)$:=1 gesetzt, andernfalls sei $a_{2s}(g)$:=0 gesetzt. – Damit sind auch die ***Werte*** aller Zusatzattribute a_{1r}, a_{2s} (r, $s \in$ **RELT$_{max}$**) wohl-definiert. Die ursprüngliche Wertemenge W zu $\mathbb{A}$ wird – falls nicht schon vorhanden – zu einer Wertemenge
$W^+ := W \cup \{0, 1\}$ zu $\mathbb{A}^+$
erweitert.

3.2 Abschließende Bemerkung

Gemäß dem Vorschlag von *Thomas Zeh* gelte nun statt (4) die Präzisierung
(4^+) $att^+(c) = att^+(d) \Leftrightarrow c = d$ für alle Klassen $c, d \in \mathbb{C}$ des O-Schemas.

Berücksichtigen wir nun noch folgende Zusatzregeln für die Klassentaxonomie $<_C$ (die auch schon in [2] erwähnt wurden):

(6) Sind c, d zwei unvergleichbare Klassen mit $\underline{c} \cap \underline{d} \neq \emptyset$, ***so besteht ein Grund***, eine (möglichst große) Klasse e einzuführen mit $\underline{e} \subseteq \underline{c} \cap \underline{d}$, so dass also in der Klassentaxonomie $e <_C c$ und $e <_C d$ folgt.
(7) Sind c, d zwei unvergleichbare Klassen mit $att^+(c) \cap att^+(d) \neq \emptyset$, ***so besteht ein Grund***, eine (möglichst große) Klasse f mit $att^+(c) \cup att^+(d) \subseteq att^+(f)$, so dass also in der Klassentaxonomie $f <_C c$ und $f <_C d$ folgt.

Mit den Voraussetzungen (1), (2), (3), (4^+), (5) erhalten wir nun wieder analog zu [2] das Ergebnis, dass die geordneten Systeme ($\mathbb{C}$, $<_C$), ([$\mathbb{A}^+$], $<_A$) in der **FBA**-Sprache durch zwei zueinander dual- isomorphe **vollständige Verbände** ($\mathfrak{U}(\mathbf{I}^+)$, $\subseteq$) („Umfangsverband"), ($\mathfrak{J}(\mathbf{I}^+)$, $\subseteq$) („Inhaltsverband") dargestellt werden können. Alles andere (Vererbung der maximalen Relationship-Types, „Verfeinerung" der Ontology durch die formalen Kontexte (c, d, r) [für relt(r) = (c, d) mit r$\in$ **RELT$_{max}$** und c, d$\in\mathbb{C}$], sowie Beschreibung der Attributwertzuweisung bei Instanziierung durch einen mehrwertigen formalen Kontext) geht analog so wie in [2] beschrieben.

Ich bin neugierig, was die Informatiker unseres ONTO-Arbeitskreises zu diesem ergänzenden Kompromissvorschlag sagen.

Mit freundlichem Gruß

Christoph Lübbert

4 Literatur

[1] *B. Ganter / R. Wille*: „Formale Begriffsanalyse" (FBA); Springer 1996.
[Die von mir am meisten benutzte Referenz]

[2] *C. Lübbert:* „Verfeinerung der Ontology-Sprache durch die FBA-Sprache" V1, Darmstadt, – Vortrag im ONTO-Kreis Darmstadt am 23.6.2014.
[Basis für die vorliegende Note]

[3] *A. Maedche & V. Zacharias:* „Clustering Ontology-based Metadata in the Semantic Web", FZI Research Center for Information Technologies at the University of Karlsruhe, Research Group WIM, 2002/2003.

[4] *C. Lübbert:* „Ontologie-Definition auf Basis von FBA", V3.14, Darmstadt, Aug. 2012
[Das ist meine frühere, ganz auf einer Instanzen- + Attribute-Menge aufbauende Ontology-Definition, die allerdings nicht wie [2] direkt den Versuch machte, zwischen Ontology-Sprache und FBA-Sprache zu *vermitteln*, sondern die konventionelle O-Definitionen wie etwa in [3] auf sie zurück zu führen]