

H. Wedekind

WHAT IS THE RIGHT OBJECT MODEL FOR A DATABASE ?

Beantwortung einer Frage von Jim Gray

Von Hartmut Wedekind (Uni Erlangen, Germany)

Abstract

In the Article there is represented Professor Hartmut Wedekind (Germany) answer for the problematic article of Professor Jim Gray and for the questions about modeling heterogeneous bases. In the article is also represented using concept of categorical analyses for the relation model schema, which includes formal grammar contractual aspects of language. The right object for a database is the semantically most precise one.

Keywords: Modeling Heterogeneous Bases, Categorical Analyses, Relation Model Schema.

Auf der SIGMOD 2004 stellte Jim Gray in seinem Vortrag „The Next Database Revolution“ eine Reihe von Forschungsfragen, die von ihm in dem folgenden Absatz zusammengefasst wurden:

“It is likely that web services will be the way we federate heterogeneous database systems. This is an active research area. **What is the right object model for a database?** What is the right way to represent information on the wire? How do schemas work in the Internet? How does schema evolution work? How do you find data and databases? We do not have good answers to any of these questions. Much of my time is devoted to trying to answer these questions for the federation of astronomy databases we call the World-Wide Telescope. “

Wir wenden uns der von uns oben herausgestellten Frage zu “What is the right object model for a database?, weil es sich um eine klassische Fragestellung handelt, auf die diverse Autoren eine Antwort zu geben versucht haben. Die für mich bedeutendste Antwort stammt von Paul Lorenzen, die er in seinem Buch „Lehrbuch der konstruktiven Wissenschaftstheorie“ (BI Verlag, 1986) gegeben hat. Ich trage Lorenzens Aufbau im wesentlichen in der Terminologie der Datenbanken und Programmierung vor. Es wird nur vorausgesetzt, dass bekannt ist, wie das Schema eines Elementarsatzes hingeschrieben wird. Das Schema gesondert herauszustellen, ist in der Informatik leider ungewöhnlich. Deshalb hier diese kurze Darstellung. Das Schema lautet: $N \varepsilon P$, mit N als Nominator (benennender Ausdruck), ε als Kopula und P als Prädikator (unterscheidender Ausdruck). „Fido ist ein Hund“ ist ein seit Carnap bekanntes Beispiel für einen Elementarsatz mit Fido als Nominator, „ist“ als Kopula, die den nominativen und prädikativen Teil eines Elementarsatzes verbindet, und Hund als Prädikator.

Für Datenbanksysteme erlangen Elementarsätze, in denen logische Junktoren ausgeschlossen sind, weil diese erst beim Aufbau komplexer Sätze herangezogen werden, im Relationenmodell eine herausragende Bedeutung. Vorausgesetzt werden muss, dass eine zweite Art von Prädikatoren, nämlich die Apprädikatoren oder Attribute (A_i), zuvor einzuführen sind. Es gibt für einen vorgegebenen Prädikator P im allgemeinen beliebig viele Apprädikatoren (A_i). Lorenzen schreibt:

$$N \varepsilon (A_1, A_2, \dots, A_n) P.$$

Im Fachgebiet Datenbanken schreibt man:

$$P(N, A_1, A_2, \dots, A_n).$$

Auffallend ist nur, dass die Kopula ε nicht explizit aufgeführt wird. Dass N Primärschlüssel heißt und P Relationen-Name genannt wird, ist nur eine terminologische, keine sachliche Verschiedenheit. $P(N, A_1, A_2, \dots, A_n)$ ist ein allgemeines konzeptionelles Schema (Metaebene) und Hund (Fido, braun, weiblich, teuer) eine Ausprägung oder Instantiierung eines speziellen Hundeschemas Hund (Name, Farbe, Geschlecht, Wert) auf der Objektebene.

Für Jim Gray, wie für viel andere auch, war das Aufkommen der objektorientierten Programmierung ein zentrales Ereignis, weil man einen Hinweis bekam, wie man das klassische Relationen - Modell um Operatoren, sprich Methoden erweitern kann. Es gab viele „Irrungen und Wirrungen“ in der Datenbankforschung. Heute im Zeitalter der Komponenten -Transaktionsmonitore à la J2EE ist das kein Problem mehr. Worum geht es?

Es muss nicht nur über Hunde, sondern auch über das Waschen von Hunden geredet werden, wobei „waschen“ ein Operator bzw. eine Methode ist. Aus der Sicht der Elementarsätze liegt ein Satz vor wie „Kunibert wäscht einen Hund“. „Von Fido ist ein brauner, weiblicher, teurer Hund“ einer relationalen Datenbank bis zu unserem Kunibert-Satz ist ein weiter Weg, weil schematisch eine ganz andere Struktur vorliegt. In „Kunibert tut (waschen) Hund“, oder „Kunibert π (waschen) Hund“ taucht eine neue Kopula

auf, die Tatkopula „ π “, um überhaupt die Methode „waschen“ einführen zu können. „Hund“ hingegen als grammatisches Objekt der Methode (ein Prädikat einer empirischen Grammatik) ist uns aus Datenbanken bekannt. Allgemein gilt für die Objektorientierung: $N \pi(Q) P$, mit Q als einem Tatprädikator (Methode). Mit beliebig vielen Methoden kann geschrieben werden: $N \pi(Q_1) P$, $N \pi(Q_2) P$, ..., $N \pi(Q_n) P$. Neben „waschen“ gibt es ja schließlich auch „kämmen“, „flöhen“ etc. Wie man die Kopula π und die Methoden Q_i technisch packt, ist bekannt. In einer Klasse „Hund“ werden die Methoden Q_1, Q_2, \dots, Q_n als Interface nach außen „sichtbar“ gemacht und „unsichtbar“ implementiert. Hier liegt ein Abstraktionsvorgang vor, auf den hier nicht eingegangen werden soll. Wer nun etwas tut, ein Kunibert oder allgemein ein N , ist aus der Sicht der Informatik uninteressant. Betrachtet werden nur die infinite Verbformen als Methode.

Jetzt kommt das von Jim geforderte Neue, das gar nicht so neu ist. In der konstruktiven Wissenschaftstheorie wird das OO-Modell ausgebaut, um zu einer reicheren Semantik zu gelangen. Elementarsätze werden erweitert um weitere, indirekte Objekte. Die Erweiterung nennt man Fallbildung. Der Mittelfall (I) wird vom Werkfall (II) (Transformationsfall) und Gebefall (III) (Transportfall) unterschieden. **Mittelfall:** Kunibert π (waschen) Hund (mit Seife). Seife ist ein indirektes Objekt. **Werkfall** (Transformationsfall): Kunibert $\pi\pi$ (verbrennen) Holz (zu Asche). **Gebefall** (Transportfall): Kunibert π (geben) Hund (an Freund). Das sind drei verschiedene Klassentypen: Class I, Class II, Class III. Die sind nicht empirisch gebildet. Sie sind rational begründet. Man orientiert sich am Mittel, um etwas zu tun. Man überführt etwas von einer Substanz in eine andere, wobei die Substanz ihre Identität verliert, und man transportiert etwas von einem Ort zum anderen. Was kann man noch tun?

Man bedenke: Bisher kannte man nur einen Klassentyp, jetzt drei.

$$N \pi(Q) P (P^I) \quad N \pi(Q) P (P^{II}) \quad N \pi(Q) P (P^{III})$$

Zusammenfassend gesehen, haben wir vier Stufen der Entwicklung:

- 1) $N \varepsilon P$ (Logischer Atomismus)
- 2) $N \varepsilon (A_1, A_2, \dots, A_n) P$ (Relationenmodell)
- 3) $N \pi(Q) P$ (OO - Modell)
- 4) $N \pi(Q) P (P^i)$ (erweitertes OO- Modell um Klassentypen i)

Rationale Grammatiken, um die es ja geht, können erweitert werden. Eine zusätzliche Erweiterung 5) gelingt über Präpositionen, die rational zu fassen sind. Wie können wir Sätze wie
 „Kunibert π (waschen) Hund (mit Seife) im Haus.“
 schematisieren?

„im“ ist eine Lokalpräposition (σ_j). Lorenzen (S.48) zählt auf, dass es rational 216 Exemplare davon gibt.

Schematisch gilt somit für den obigen Satz:

- 5) $N \pi(Q) P_1 (P^i) \sigma_j P_2$. (erweitertes OO- Modell um Präpositionen)

Danach gibt es formal für $i=3$ und $j=216$ insgesamt $3 \times 216 = 648$ Klassentypen. Durch logische Junktoren kann man natürlich eine beliebige Vielzahl komplexer Klassenarten erzeugen, was aber uninteressant ist.

Die Graysche Frage, anders wiederholt, lautet:

Welche Klassentypen brauchen wir? Das hängt von den benötigten Präpositionen ab. Vielleicht kann man sich die an einer Hand abzählen, d.h. wörtlich genommen $3 \times 5 = 15$. Das wäre schön für eine Normierung und Standardisierung.

„Wozu der Lärm“, fragt Mephisto in der dann beginnenden Paktszene im Faust. „Das also war des Pudels Kern, ein fahrender Scholast? Der Kasus macht mich lachen“ antwortet Faust.

Wo liegt nach dem Lärm des Pudels Kern in unserem Falle. Ganz einfach: Wir verfügen mit den Fall- und präpositionalen Erweiterungen über eine höhere Modelliergenauigkeit, das ist alles! Mehr nicht. Das kann aber gewaltig sein!

Gehen wir in die Geschichte der Logik, so ist der Übergang vom Elementarsatz 1) $N \varepsilon P$ zum Relationen-Modell 2) $N \varepsilon (A_1, A_2, \dots, A_n) P$ auch ein Schritt hin zu höherer Genauigkeit. Denn: $N \varepsilon P$, das ist Logischer Atomismus aus den Zeiten eines Bertrand Russell (1872-1970). Aus „Fido ist eine brauner Hund“ wurde zu diesen Zeiten ohne Apprädikatoren: Fido ε Hund UND Fido ε braun. Eine Unterscheidung zum: „Fido ist ein hündisches Braun“ ist nicht mehr möglich, weil eben die Rekonstruktion nur einem Prädikator kennt, um den sich alles dreht, wie die Elektronen um einen Kern (deshalb die Metapher „Logischer Atomismus“).

Dann kam das Relationenmodell (1970) und man verwies das Adjektiv „braun“ in den Bereich der Apprädikatoren oder Attribute (A_i). Aus dieser Sicht ist „Fido ist ein brauner Hund“ immer noch ein Elementarsatz, nämlich „Fido ε (braun) Hund, der vom „Fido ε (hündisch) Braun“ deutlich verschieden ist.

Das ist ja auch beabsichtigt bei unserem Streben nach höherer Genauigkeit. Denn: Beide Sätze „Fido ist eine brauner Hund“ und „Fido ist ein hündisches Braun“ ununterscheidbar zu machen ist – vornehm gesprochen - ein starkes Stück, das mit zum schlechten Ruf der Logik in der Welt beigetragen hat. Goethe in Dichtung und Wahrheit über seine Leipziger Studienzeit: „In der Logik kam es mir wunderbarlich vor, dass ich diejenigen Geistesoperationen, die ich von Jugend auf mit der größten Bequemlichkeit verrichtete, so auseinanderzerren, vereinzeln und gleichsam zerstören sollte, um den rechten Gebrauch derselben einzusehen.“

Der Übergang vom Relationen-Modell $2) N \varepsilon (A_1, A_2, \dots, A_n) P$ zum OO -Modell $3) N \pi (Q)P$ war gewaltig. Zu vermerken ist an dieser Stelle noch, dass natürlich auch die Tatprädikatoren oder Methoden Q Apprädikatoren kennen, die man empirisch-grammatisch Adverbien (Umstandswörter) bzw. programmtechnisch Argumente nennt. „Kunibert π (schnell) (waschen) Hund“ ist ein gültiger Satz des Schemas:

$N \pi (B_1, B_2, \dots, B_m) (Q) P$. Die B_i heißen Formalparameter (Argumente) der Methode Q .

Das nur als Vorbemerkung bevor wir jetzt weitere Ungenauigkeiten aufdecken.

Wir beginnen aus Gründen der Einfachheit mit „präpositionalen Ungenauigkeiten“. Wie modelliert man: „Fido ist ein brauner Hund in Köln“ oder „Fido ε (braun) Hund in Köln“. Antwort: Da man Präpositionen nicht kennt, schiebt man den präpositionalen Teil in den Bereich der Apprädikatoren und macht daraus: „Fido ist ein brauner, kölnischer Hund“ als Ausprägung eines Schemas „Name ε (Farbe, Ort) Hund“. Zwischen „Hund in Köln“ und „kölnischer Hund“ ist aber ein gewaltiger Unterschied, der relational wegeskamotiert wird, ein ähnlicher Trick, den auch die logischen Atomisten“ vor 1970 mit ihrer Gleichstellung von „braunem Hund“ und „hündischem Braun“ angewandt haben.

Nun zu Ungenauigkeiten durch eine nicht vorhandene Fallbildung im Relationen - Modell und OO-Modell:

Aus „Kunibert π (schnell)(waschen) Hund (mit Seife)“ werden, da man den wichtigen Mittelfall einer allgemeinen Zweck-Mittelrelation nicht kennt, die Mittel dem Apprädikatorenbereich (Argumentbereich) einer Methode zugeschrieben. Man spricht jetzt von einem „seifigen Waschen“ und schreibt:

Kunibert π (schnell, seifiges) (waschen) Hund.

Es ist heute üblich, Fallunterscheidungen und präpositionale Erweiterungen den Methodenparametern und Attributen zuzuweisen. Man löst alles apprädikativ. Die semantischen Verbiegungen, die damit eingeführt werden, sind seit den Zeiten der Überwindung des Logischen Atomismus bekannt.

Wie kann man die Erweiterungen des OO-Modells im Sinne von 4) und 5) in der Sprache der Informatik formulieren? Dazu gehen wir von einem „gewichtigeren“, technischen Beispiel aus:

$N \pi$ (verschrauben) Deckel (mit Werkzeug)

Die Methode „verschrauben“ mag in einem **interface** *Montage* { void verschrauben (); .. } der Klasse „**Class** Deckel“ aufgeführt werden. Die Methode bedarf aber zur Ausführung der indirekten Klasse „**Class** Werkzeug“ mit dem **interface** *Zugriff* { get_tool (); ..}. Eine Mittelrelation „with“ z.B. zwischen **Class** Deckel und **Class** Werkzeug wird von OO-Sprachen nicht unterstützt. Gleiches gilt für alle anderen, vorgeschlagenen Erweiterungen auch. OO-Sprachen unterstützen nur die Relation „Class-Subclass“. Es wäre eine schlechte Idee, zum Instrument der Spracherweiterung zu greifen. Spracherweiterungen lassen sich nicht durchsetzen. Sie sind zu aufwendig. Besser ist es, auf Metaebene eine Klassen-Auswahlsprache vorzusehen, die etwa als Skriptsprache ausgebildet werden kann.

Um eine Mittelrelation zwischen einer direkten und indirekten Klasse zu spezifizieren, genügt dann z.B. der folgende Skript-Eintrag:

„Choose_direct **Class** Deckel , Choose_indirect **Class** Werkzeug, Case: with;

Wenn jetzt geschrieben wird,

```
Class Deckel {
    void verschrauben ( ) {...}
    Class Werkzeug {
        get_tool ( ) {...}
    }
}
```

dann ist die meta-sprachlich fixierte Semantik des Mittelfalles objekt-sprachlich (programm-sprachlich) verbindlich. In ähnlicher Weise ist mit den anderen beiden Fällen und einer präpositionalen Ergänzung zu verfahren.

Die Bedeutung der Semantik der vorgetragenen OO-Erweiterung liegt in der rationalen Begrenzung auf die Fallzahl 3 und einer beschränkten Zahl von Präpositionen. Gegenüber ein schierer Unbegrenztheit rein empirischer Erweiterungen führt diese rationale Reduktion zu einer semantischen Beherrschbarkeit und Absicherung gegen Willkür, was auch beabsichtigt ist.

Wie lautet die Antwort auf Jim Grays Frage: „What is he right object model for databases ?“. Ganz einfach: “The semantically most precise one is the right one”.

References

1. Wedekind H. Objectorientierte Schemaentwicklung. Ein kategorialer Ansatz fuer Datenbanken und Programmierung. BI-Wiss.-Verlag, Mannheim/Wien/Zuerich., 1992.
2. Lorenzen P. Lehrbuch der konstruktiven Wissenschaftstheorie. Logische. BI-Wiss.-Verlag, Mannheim, 1987.

ჰ. ვედეკინდი (გერმანია)

**რომელი ობიექტური მოდელია სწორი მონაცემთა ბაზისათვის ?
(ჰარტმუტ ვედეკინდის პასუხი ჯიმ გრეის შეკითხვაზე)**

რეზიუმე

სტატიაში შემოთავაზებულია პროფესორ ჰარტმუტ ვედეკინდის (გერმანია) პასუხი პროფესორ ჯიმ გრეის პრობლემატურ სტატიაზე და შეკითხვებზე მონაცემთა არაერთგვაროვანი (ჰეტეროგენული) ბაზების მოდელირების შესახებ.

სტატიაში მოცემულია მონაცემთა რელაციური მოდელის სქემისთვის კატეგორიული ანალიზის გამოყენების კონცეფცია, რომელიც ენის ფორმალური გრამატიკის კონსტრუქციულ ასპექტებს მოიცავს. მონაცემთა ბაზისათვის ის ობიექტური მოდელია სწორი, რომელიც სემანტიკურად ყველაზე სწორია.

X. ВЕДЕКИНД (Германия)

**КАКАЯ ОБЪЕКТНАЯ МОДЕЛЬ ЯВЛЯЕТСЯ ТОЧНОЙ ДЛЯ БАЗЫ ДАННЫХ ?
(Ответ Хартмут Ведыкина на вопрос Джим Грея)**

Резюме

В статье предлагается ответ профессора Хартмута Ведыкина (Германия) на проблемную статью и вопросы профессора Джим Грея (США) о моделировании неоднородных (гетерогенных) баз данных.

В статье изложена концепция использования категориального анализа для реляционной модели данных, которая охватывает конструктивные аспекты формальной грамматики языка. Для баз данных та объектная модель является точной, которая семантически является самой правильной.

