

INTERLIS 2 zu 1: Modell- und Typen-Abbildungen



Prof. Stefan F. Keller, sfkeller@hsr.ch

Ausgabe 3 vom 2002-10-29 (deutsch)

Inhaltsverzeichnis

1	Einführung.....	2
1.1	Zweck und Inhalt dieses Dokuments.....	2
1.2	Weitere Informationen.....	2
1.3	Grundsätzliches.....	2
2	Abbildung von OO- zu ER-Datenmodellen.....	2
2.1	Vererbung.....	2
2.2	Beziehungen.....	5
2.3	Unterstrukturen.....	5
3	Abbildung von INTERLIS 2- zu 1-Datentypen	6
	Anhang: Quellenangaben und Versionenverzeichnis	7

1 Einführung

1.1 Zweck und Inhalt dieses Dokuments

Dieses Dokument ist ein („work-in-progress“) über aktuelle Fragen der Ableitung und -Zuordnung von Modellen, z.B. von konzeptionell-externen Schemas zu internen Schemas.

Konkret gehören dazu auch Fragen, wie das „Mapping“ von INTERLIS 2 zu INTERLIS 1 oder von objekt-orientierten zu relationalen Datenmodellen.

1.2 Weitere Informationen

Die Hauptinformationsquellen sind:

- <http://www.INTERLIS.ch>, und
- <http://www.integis.ch>

Siehe auch.

- Das INTERLIS 2-Referenzhandbuch
- Benutzerhandbuch zum INTERLIS 2-Compiler

Kontakt: siehe Titelblatt.

Der Autor (bzw. die Autorinnen und Autoren) sind damit einverstanden, dass Veröffentlichungen und Vervielfältigungen ihrer Beiträge erfolgen und dass die Urheber- und Nutzungsrechte aller Beiträge den Herausgebern der INTERLIS-Dokumente zu gleichen Rechten übertragen werden.

1.3 Grundsätzliches

Die unterschiedlichen Betrachtungsweisen von INTERLIS 1 und INTERLIS 2 führen zu einigen grundsätzlichen Abbildungsproblemen. Sie werden im Kapitel 2 beleuchtet.

Es sollen wenn imm möglich Alternativen aufgezeigt werden, mit denen einige vorhersehbare Probleme im Zusammenhang mit den bestehenden Systemen und Datenbeständen vermieden werden können.

Es ist zu Unterscheiden zwischen folgenden Abbildungen von INTERLIS 2 nach INTERLIS 1-Modellen:

- Abbildung von OO- zu ER-Datenmodellen
- Abbildungen der Datentypen von INTERLIS 2 zu 1

2 Abbildung von OO- zu ER-Datenmodellen

2.1 Vererbung

Text auf Basis Dorfschmid (2000).

2.1.1 Abstrakte Klassen

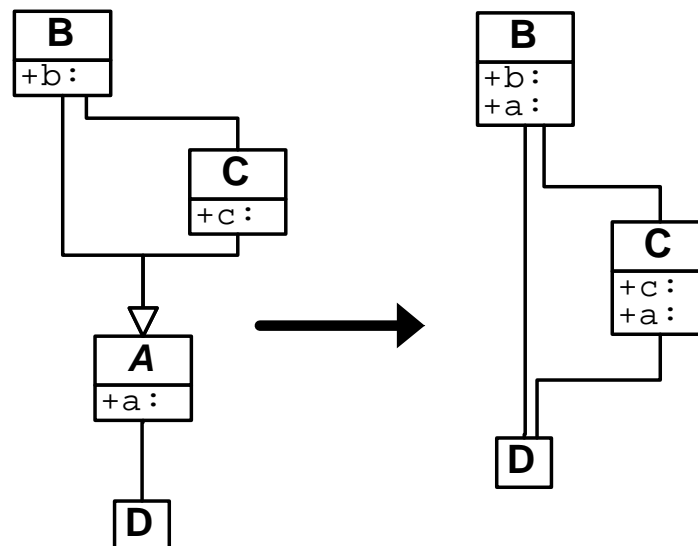
Siehe Klassen.

2.1.2 Klassen

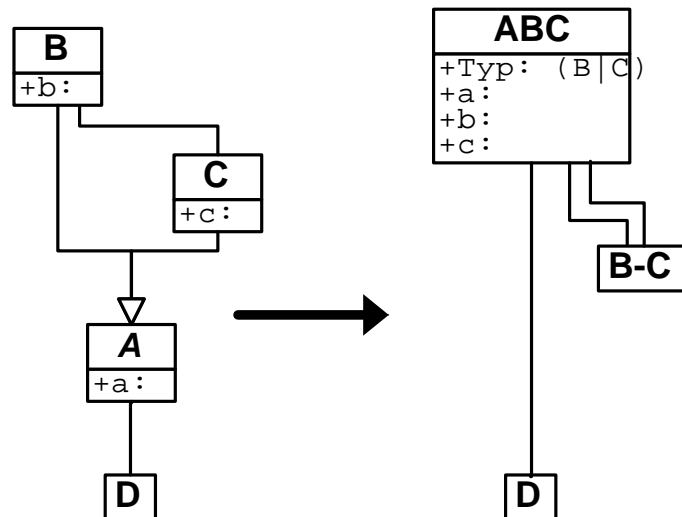
In vielen OO-Datenmodellen werden zunächst abstrakte Objektklassen (Klasse A im Beispiel) eingeführt und diese dann zu weiteren abstrakten und schliesslich zu konkreten Klassen (Klassen B und C im Beispiel) spezialisiert. Dieses Mittel ist typisch für den objektorientierten Entwurf. Um das Beispiel realistischer zu gestalten, gibt es eine Beziehung zwischen den Subklassen B und C und es wird noch eine konkrete Klasse D eingeführt, die eine Assoziation zur Klasse A hat.

Für die Abbildung in ein relationales Modell stehen verschiedene Möglichkeiten zur Verfügung:

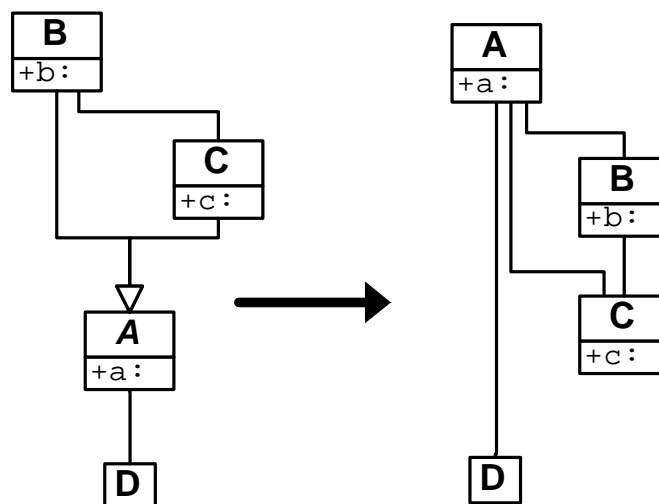
1. Jede konkrete Objektklasse (B, C, D) wird in eine relationale Tabelle abgebildet (Anschaulich: Die Attribute der Oberklasse "wandern" zu den Unterlassen). Diese enthält dann alle Attribute der entsprechenden konkreten Objektklasse und aller durch sie geerbten Superklassen. Diese Möglichkeit entspricht am ehesten einer natürlichen Auffassung. Sie hat aber den Nachteil, dass Beziehungsattribute, die auf eine abstrakte Klasse verweisen auf verschiedene konkrete Tabellen verweisen müssen. Dies ist in der relationalen Methode nur mittels mehrerer Beziehungsattribute (im Beispiel je ein Beziehungsattribut von D zu B bzw. von D zu C) oder mit zusätzlichen Beziehungstabellen beschreibbar.



2. Die Summe aller aus einer Basisklasse spezialisierten Klassen wird auf eine einzige relationale Tabelle (Tabelle ABC) abgebildet (Anschaulich: Die Attribute der Unterklassen "wandern" zur Oberklasse). Diese Tabelle enthält Attribute, die in den verschiedenen Klassendefinitionen vorkommen und zusätzlich ein Attribut, das beschreibt, zu welcher konkreten Objektklasse ein Objekt gehört. In einem konkreten Objekt sind dann dieser Objekttyp und diejenigen Attribute definiert, die für die jeweilige Spezialisierung in Frage kommen. Diese Abbildungsweise kann zwar etwas unübersichtlich sein; sie hat aber den Vorteil, dass keine Schwierigkeiten mit Beziehungsattributen entstehen, die von Drittobjekten ausgehen. Beziehungen, die innerhalb von Objekten bestehen, die in der gleichen relationalen Tabelle abgebildet werden, müssen allerdings auch mittels einer Beziehungstabelle dargestellt werden, damit die INTERLIS 1-Vorschrift eingehalten werden kann, wonach nur schon bekannte Objekte gezeigt werden dürfen. Zudem müssen alle obligatorischen Attribute von Subklassen in der Tabelle auf nicht-obligatorisch geändert werden.



3. Für jede abstrakte oder konkrete Objektklasse wird eine entsprechende relationale Tabelle gebildet (Anschaulich: Klassen werden 1:1 zu Tabellen abgebildet). Die Tabellen enthalten jeweils die Attribute gemäss der Objektklasse. Spezialisierende Klassen enthalten zusätzlich einen Verweis (Beziehungsattribut) auf die jeweilige Superklasse (Im Beispiel also auf A). Ein konkretes Objekt wird entsprechend der Vererbungsstruktur in mehrere Teilobjekte zerhackt, die jeweils auf das zugehörige Superobjekt verweisen. Dazu kommt bei der Basisklasse ein diskriminierendes Attribut `_Typ`, das hilft, die Subklassen zu unterscheiden (kann evtl. weggelassen werden). Dieses Vorgehen hat den Vorteil, dass keinerlei Probleme mit Beziehungen entstehen. Es hat aber den Nachteil, dass die Objekte in Teilobjekte zerhackt werden. Die damit verbundene grössere Objektmenge ist kaum von Belang. Es muss aber damit gerechnet werden, dass die Erstellung der zerhackten Objekte und vor allem ihre Zusammenfügung in verschiedenen Systemen zu Problemen führen könnte.



Wir empfehlen Variante 3 (Klassen werden 1:1 zu Tabellen abgebildet).

Hinweise:

- Views für Subklassen-Objekte: Ein interessante Überlegung ist, ob die "OO-zu-relational"-Abbildung

(= ili2-to-pgsql-CREATE TABLE-Software) nicht gleich Views für die Subklassen definieren könnte, so könnte man Subklassen-Objekte einfacher abfragen.

2.1.3 Vererbung von EXTENDED-Attributen:

EXTENDED-Attribute in Subklassen (typischerweise erweiterte Aufzähltypen) werden auch in den Subklassen aufgeführt mit dem Unterschied, dass dort die erweiterten Werte und gleichzeitig (d.h. anteilmässig redundant) in der Basisklasse die ursprünglichen Werte stehen (ich glaube, die Alternative, die das Attribut mit den erweiterten Werten nur in der Subklasse zu verwalten, ist nicht machbar, denn sonst könnte es keine konsistenten Objekte gemäss Basis mehr geben).

Beispiel: Siehe wieder das Roads-Beispiel: Das

```
CLASS PointObject( Type, Position );
CLASS PointObjectExtended EXTENDS PointObject ( Type(EXTENDED) );
```

... wird relational zu (siehe _Type und REF):

```
SQL-TABLE PointObject( _TID, Type, Position, _Type );
SQL-TABLE PointObjectExtended ( _TID, Type, _Ref: FOREIGN KEY REFERENCES
PointObject );
```

Anderes Beispiel: Gegeben, in einer Basisklasse gibt es den Aufzählungswert "rot" und eine die Basisklasse erweiternde Subklasse erweitert diesen Attributtyp mit "rot.hell"; dann würde ein Tupel der "Basisklasse-Tabelle" (als Teil des Subklassen-"Gesamtobjekts") den Wert "rot" erhalten und ein Tupel der "Subklasse-Tabelle" würde den Wert "rot.hell" erhalten.

2.2 Beziehungen

2.2.1 Assoziationen

Im Falle von "zweiwertigen" Assoziationsklassen ohne zusätzlichen Attributen, bei denen eine der Rollen die Kardinalität 1 hat, kann diese weggelassen werden, indem die Rolle dem anderen bezeichneten Tabelle zugeordnet wird.

In den anderen Fällen, ist eine Tabelle zu schaffen ("Kreuz- oder Zwischen-Tabelle).

2.2.2 Beziehungen über Themen(bzw. Behälter-)Grenzen hinweg

Müssen als Primär-Fremdschlüssel abgebildet werden.

2.2.3 Aggregation und Komposition von Klassen

Aggregationen werden 1:1 in Tabellen abgebildet.

Kompositionen in INTERLIS 2 müssen in INTERLIS 1 als eigenständige Objekte aufgefasst werden, die jeweils eine Beziehung zum "Oberobjekt" aufweisen. Wird dieselbe Struktur in verschiedenen Objekten als Komposition verwendet, müssen jeweils verschiedene Tabellen gebildet werden, da die Beziehung jeweils auf verschiedene Oberobjekte verweist.

Siehe auch Kapitel 2.3.1 Kompositionsattribute mit LIST/BAG OF.

2.3 Unterstrukturen

2.3.1 Kompositionsattribute mit LIST/BAG OF

Attributtypen, die in INTERLIS 2 mit BAG OF (Objektreihenfolge unbestimmt, bzw. LIST OF (Reihenfolge relevant) definiert sind, werden in eine separate, zusätzliche Tabelle ausgelagert, die eine INTERLIS 1-Beziehung auf die ursprüngliche Tabelle erhalten. Bei LIST OF ist in der zusätzlichen Tabelle nebst der Beziehung auch eine Ordnungsnummer zu führen.

3 Abbildung von INTERLIS 2- zu 1-Datentypen

Folgende Zuordnungen werden gemacht:

- URI => TEXT
- DIRECTED POLYLINE => POLYLINE (DIRECTED wird ignoriert)
- INTERLIS_1_DATE => DATE;
- BOOLEAN => (false, true) !! ergibt die Werte (0, 1)

Für weitere Angaben siehe das INTERLIS 2-Compiler-Handbuch.

Man beachte, dass die INTERLIS 2 zu 1-Option gemeinsame DOMAINS nur noch einzeln bei den Attributen angibt, obschon alle auf einen gemeinsamen Domain zurückführen. Steht also in INTERLIS 2 "DOMAIN LKoord = COORD..." so findet man diesen Domain im INTERLIS 1-Text nicht mehr: Der Attribut-Typ wird als "COORD2..." allen Geometritypen direkt angehängt. Ich würde erwarten, dass diese Information erhalten bleibt, d.h. das "DOMAIN LKoord = COORD2..." auch im INTERLIS 1-Modell erscheint. Das hat keinen direkten Einfluss auf die Daten, nur auf die Semantik der Beschreibung.

Anhang: Quellenangaben und Versionenverzeichnis

Quellenangaben

Dorfschmid (2000): Expertise "Beurteilung des Thema Gebäudeadressen des Datenmodells 01 der amtlichen Vermessung" (29. Dezember 2000) von J. Dorfschmid, ADASYS AG, Kronenstrasse 38, 8006 Zürich, do@adasys.ch, www.adasys.ch.

Versionenverzeichnis

Ver.	Dateiname	Datum	Wer	Was?
01	Schema-Mapping_r01.doc	2001-10-01 und 2001-09-25	SFK	erstellt (Dateiname war: Glossar_Addendum?), Titel war: "Modell-zu-Modell Abbildungen (INTERLIS 2-zu-1)"
02	Schema-Mapping_r01.doc	2001-11-22	SFK	Kategorien eingeführt, neuer Titel
03	MM_ILI2-zu-1_r03.doc	2002-10-29	SFK	Erweitert

□