
Modellieren raumbezogener Daten

**Eine Einführung unter Berücksichtigung
von UML und INTERLIS**

(auf INTERLIS 2.3 angepasste und ergänzte Ausgabe)

Herausgeber

Koordination, Geo-Information und Services (KOGIS)
c/o swisstopo, Bundesamt für Landestopografie
Seftigenstrasse 264, CH – 3084 Wabern

www.kogis.ch

Autoren

Joseph Dorfschmid · do@adasys.ch

Sascha Brawer · sb@adasys.ch

Adasys AG, Dörflistrasse 67, CH–8050 Zürich

www.adasys.ch

Dank

Die Autoren danken allen, die mit Anregungen, Kritik und präziser Lektüre zu diesem Papier beigetragen haben.

Inhalt

Inhalt	3
1. Lektüre zwischen Dichtung und Wahrheit	7
2. Überblick am Beispiel Ilistal	9
2.1 Aufbruchstimmung im Ilistal	9
2.1.1 Der Startschuss	9
2.1.2 Erste Auslegeordnung	9
2.2 Erste Gehversuche	10
2.2.1 Der nationale Tourismusverband hat Vorarbeit geleistet	10
2.2.2 Wie viele Bahnen betreibt eine Bahngesellschaft?	13
2.2.3 Bergbahnen, Bahngesellschaften und Abonnements haben Eigenschaften	13
2.2.4 Modelle? Ilistal will Daten!	15
2.2.5 Ilistal sendet	16
2.3 Ilistal will mehr	16
2.3.1 Zielsetzung	16
2.3.2 Ilistal nutzt Vorhandenes	17
2.3.3 Ilistal geht weiter als der nationale Verband	18
2.3.4 Ilistaler Spezialitäten	20
2.3.5 Wie implementieren die Ilistaler ihre Spezialitäten?	21
2.3.6 Wie schicken die Ilistaler ihre Daten an den nationalen Tourismusverband?	22
2.3.7 Was macht der nationale Tourismusverband mit den Ilistaler Daten?	24
2.4 Ilistal hat es geschafft	24
2.4.1 Systemübersicht	24
2.4.2 Für die Web-Seite ist nur der aktuelle Zustand interessant	25
2.4.3 Hotels mit freien Betten auf der Web-Seite darstellen	25
3. Möglichkeiten zur Beschreibung von Daten	27
3.1 Beschreibung als Grafik: Unified Modeling Language UML	27
3.2 Beschreibung als Text: INTERLIS	27
3.3 Standardisierte Datenmodelle	29
3.4 Formate zum Datenaustausch	29
3.5 Wichtige Informationsquellen	30
4. Das Beispiel Ilistal konkret	31
4.1 Das Datenmodell im Überblick	31
4.2 Das Datenmodell in INTERLIS	32
4.2.1 Ahland.ili – Das nationale Grundlagenmodell	32

4.2.2 Adressen.ili – Das Modell für Gebäudeadressen	33
4.2.3 NatTour.ili – Das Modell des Nationalen Tourismus-Verbands	34
4.2.4 IliTour.ili – Das Ilistaler Tourismus-Modell	37
4.3 Transferdaten	40
5. Erben wird Mode.....	43
5.1 Erbstücke und Erbrecht – Prinzipien der Vererbung	43
5.2 Ererbtes verfeinern	45
5.3 Gibt es das denn überhaupt? – Abstrakte Klassen	46
5.4 So genau wollen wir das nicht vorschreiben – Abstrakte Eigenschaften	46
5.5 Details interessieren nicht – Das Spezielle allgemeiner betrachten	47
5.6 Vererbung im Grossen.....	47
5.7 Einfach- und Mehrfachvererbung	49
6. Das Datenmodell unter der Lupe	51
6.1 Taler und Groschen – Numerische Datentypen	51
6.1.1 Wertebereich	51
6.1.2 Einheiten	52
6.1.3 Numerische Typen erben	52
6.1.4 Schranken noch unbekannt.....	53
6.2 Arten von Bergbahnen – Modellierung von Arten von Objekten	53
6.3 Gibt es auch hellblaue Skipisten? – Strukturierte Aufzählungen.....	55
6.3.1 Gewöhnliche Aufzählungen und ihr Erbrecht	55
6.3.2 Unteraufzählungen	56
6.4 Ilistaler halten sich kurz – Zeichenketten und ihre ErbregeIn.....	57
6.5 Windstille – Fakultative und obligatorische Attribute	57
6.6 Wartezeiten und Fahrzeiten – Wertebereiche	58
6.7 Wo liegt das Ilistal? – Koordinatentypen	58
6.7.1 Grundsätzliches zu Koordinatentypen.....	58
6.7.2 Die umwickelte Zwetsche – Was ist ein Koordinatensystem?	59
6.7.3 Angaben zum Koordinatensystem – Metadaten	60
6.7.4 Verschiedene Koordinatensysteme	61
6.7.5 Dreidimensionale Koordinaten	62
6.8 Ist Null im Norden? – Festlegungen für Winkel und Richtungen	64
6.9 Ist eine Piste eine Linie oder eine Fläche? – Geometrietypen.....	65
6.9.1 Einfache konzeptuelle Sicht einer Linie.....	65
6.9.2 Liniestücke.....	66
6.9.3 Gerichtete Linien	67
6.9.4 Flächen	67
6.9.5 Dreidimensionale Linientypen	69
6.10 Wie bläst der Wind? – Strukturen.....	70
6.10.1 Mehrgliedrige Eigenschaften	70
6.10.2 Mehrere Strukturelemente	71
6.10.3 Strukturen und Klassen	71
6.10.4 Linien sind spezielle Strukturen.....	72
6.11 Wie wird im Ilistal gesprochen? – Mehrsprachigkeit.....	72
6.11.1 Pro Sprache ein Attribut	72

6.11.2 Sprachabhängige Bezeichnungen als Strukturelemente	73
6.11.3 Strukturelemente dürfen auf Objekte verweisen	74
6.12 Wie ticken die Ilistaler? – Modellierung von Zeit	74
6.12.1 Für einfache Ansprüche genügend	74
6.12.2 Zeitdauer als Struktur	75
6.12.3 Genaue Zeitdauer	76
6.12.4 Formatierte Darstellung von Strukturen	77
6.12.5 Zeitpunkte	77
6.13 Tarifbereiche, Zustandsmeldungen – Beziehungen	78
6.13.1 Rollen	78
6.13.2 Stärke einer Beziehung	79
6.13.3 Beziehungen mit Attributen	80
6.13.4 Mehrgliedrige Beziehungen	81
6.13.5 Geordnete Beziehungen	82
6.13.6 Beziehungen erweitern	83
6.13.7 Ableitbare Beziehungen	85
6.14 Einzigartige Ilisthornbahnen – Konsistenzbedingungen	86
6.14.1 Grundsätzliches	86
6.14.2 Plausibilitätsbedingungen	87
6.14.3 Eindeutigkeitsbedingungen	88
6.14.4 Existenzfordernis	90
6.14.5 Vererbung von Konsistenzbedingungen	91
6.15 Wie eng gehören Betriebsentscheide zur Bahn? – Unabhängige Themen	91
6.15.1 Allgemeines	91
6.15.2 Unabhängigkeit von Themen	92
6.15.3 Die Verantwortung von Sender und Empfänger	92
6.16 Alles Gute kommt von oben – Vorhandene Modelle nutzen	93
6.17 Tarifbereiche interessieren nicht – Sichten	93
6.17.1 Allgemeines	93
6.17.2 Das Bildungsgesetz von Sichten	94
6.17.3 Schrittweiser Aufbau von Sichten	97
6.17.4 Sichten erben	98
6.18 Namen sind Schall und Rauch – Schema in einer Fremdsprache	99
7. Die Ilistaler Systeme im Blickfeld	101
7.1 Was sind normgerechte Systeme? – Systemneutralität	101
7.2 Der Eurokurs schwankt von Tag zu Tag – Parameter und Funktionen	102
7.3 Auf krummen Wegen – Linienformen	103
8. Die Ilistaler Daten unterwegs	105
8.1 Aus den Augen, aus dem Sinn – Volltransfer	105
8.2 Klammern so spitz wie das Ilisthorn – Transferregeln auf Basis XML	105
8.3 Einmal und immer wieder – Inkrementelle Nachlieferung	107
8.4 Auch die Blauberge sind touristisch – Behälter, Replikate, polymorphes Lesen	107
8.5 Der Ponylift im «Val de la marmotte jaune» – Fremdsprachen beim Datentransfer	109
9. Datenmodellierung über das Ilistal hinaus	111
9.1 Viele Wege führen aufs Ilisthorn – Das richtige Modell gibt es nicht	111

9.2 Das Ilishorn aus verschiedenen Blickwinkeln – Originaldaten und Sichten	111
9.3 Nagt der Zahn der Zeit am Ilishorn? – Lebenszyklen	112
9.4 Vor lauter Bäumen den Wald nicht mehr sehen – Detaillierungsgrad.....	113
9.5 Das Kurorchester Ilisbad spielt auf – Datenmodellierung ist anspruchsvoll	113
10. Schlussbericht für Behörden und Management	115
Register	119

1. Lektüre zwischen Dichtung und Wahrheit

Wer kennt ihn nicht, diesen Eindruck: Das verstehe ich doch nicht! Und all diese Fachwörter! Muss das wirklich so kompliziert sein?

«Modellieren raumbezogener Daten» nähert sich darum dem Thema aus Laiensicht und bettet das Ganze in eine Geschichte ein: Die Erfolgsstory von Ilistal. Ilistal ist nirgendwo, weil es erfunden ist. Ilistal ist aber gleichzeitig überall, weil die Geschichte typische Situationen und Fragestellungen aufnimmt, die wir alle manchmal nur zu gut kennen. In diesem Dokument liegt Ilistal in Ahland, das ebenfalls erfunden ist. Dieses Hin und Her zwischen der erdichteten Welt des Ilistals und derjenigen der harten Wahrheiten von Theorie und Praxis findet mehrmals statt, für verschiedene Stufen des Interesses und des Kenntnisstandes.

- ◆ **Überblick** – Der Überblick im Kapitel 2 führt die wichtigsten Elemente der Datenmodellierung und des modellbasierten Datentransfers ein.
- ◆ **Modellierungsmethoden** – Kapitel 3 beleuchtet verschiedene bekannte Modellierungsmethoden, zeigt auf, wo UML und INTERLIS positioniert sind, und wo zusätzliche Informationen gefunden werden können.
- ◆ **Ilistaler Beispiel** – Im Kapitel 4 wird das Ilistaler Beispiel detaillierter ausgeführt. Dabei werden mögliche Anliegen im Bereich des Tourismus modelliert. Es wird dabei aber nicht der Anspruch erhoben, eine sachgerechte Lösung zu definieren. Das Hauptaugenmerk liegt darin, Modellierungsprinzipien an Hand von Beispielen aufzuzeigen. Verweise auf zusätzliche Erklärungen in den Kapiteln 5 bis 8 erleichtern den Zugang zur Theorie. Für diejenigen, die bereits einiges über Datenmodellierung und modellbasierten Transfer wissen, ist dieses Kapitel vermutlich der ideale Einstieg.
- ◆ **Diskussion des Beispiels** – Da die Vererbung ein zentrales Prinzip der vorgestellten Modellierungsmethode ist, befasst sich Kapitel 5 näher damit. Die folgenden Kapitel behandeln die Datenmodellierung im Allgemeinen (Kapitel 6), die Auswirkungen auf die eingesetzten Softwarepakete (Kapitel 7) und den Datentransfer (Kapitel 8). Wem das zu sehr ins Detail geht, der lässt Kapitel 5 – 8 einfach weg. Diese Kapitel können auch punktuell gelesen werden. Die Titel der Unterkapitel bestehen darum immer aus einem Teil gemäss der Ilistaler Geschichte und einem Fachausdruck.
- ◆ **Grundsätzliche Überlegungen** – Kapitel 9 greift einige grundsätzliche Aspekte auf, die beim Modellieren beachtet werden sollten. Die Lektüre dürfte daher auch für jene von Interesse sein, die sich weniger für die Details interessieren.
- ◆ **Schlussbericht für Behörden und Management** – Das abschliessende Kapitel 10 fasst nochmals die wichtigsten Punkte (zuhanden der Ilistaler Behörden und des Managements von Bahnen und Tourismusorganisation) zusammen. Der gehetzte Manager der realen Welt springt am besten gleich zu diesem Kapitel und verschafft sich damit einen Einstieg in die Problematik und die Möglichkeiten.

2. Überblick am Beispiel Ilistal

2.1 Aufbruchstimmung im Ilistal

2.1.1 Der Startschuss

Die Ferienregion Ilistal hat sich entschlossen, ihre Webpage auf Vordermann zu bringen. Das vielfältige Transportangebot soll interaktiv mittels eines grafischen Dialogs abrufbar sein. Die Präsentation im Gemeindehaus war beeindruckend: Schöne Bilder, trendige Abkürzungen wie HTML, XML, GIS, SVG! Sogleich kamen natürlich weitere Wünsche auf. Im Bauamt entschloss man sich kürzlich, die Adressen nach der neuen Norm zu erfassen. Das könnte man doch nutzen! Der Direktor der Ilishornbahnen erinnert sich, dass der nationale Bergbahnenverband einen Dienst einrichten möchte, über den das landesweite Bergbahnangebot verfügbar ist. Dabei sollen auch die Fahrpreise und insbesondere die verschiedenen Verbundabonnemente abrufbar sein. Sein Kollege, der für die technischen Belange zuständig ist, macht darauf aufmerksam, dass er auf seinem Compi die ganze Infrastruktur verwalte. Informationen über die Bahntrassees und die Pisten könne er schon liefern. Da sei aber nicht alles klar. Kürzlich habe er nämlich auf dem Bauamt gefragt, ob er nicht die Daten der neuen Überbauung «Im Oberboden» haben könne. Sie hätten es dann schliesslich hingekriegt. Einige Informationen seien zwar verloren gegangen. Die seien aber nicht so wichtig gewesen.

Da wird die Gemeindepräsidentin hellhörig. Eine Kollegin, die in einer Gemeinde nahe der Hauptstadt tätig ist, hat ihr doch erzählt, dass sie in einem Teilbereich bereits die dritte Software im Einsatz hätten. Zusatzwünsche, die sie eigentlich normal fand, führten jeweils zur vollständigen Überarbeitung. Sie hätten dann Unterstützung von einem Informatiker erhalten, der weniger von den verschiedenen Techniken sprach. Vielmehr habe man zuerst gemeinsam überlegt, welche Daten mit den Problemen verbunden seien. Seitdem man so vorgehe, sei es auch viel besser möglich, schrittweise Erfolge zu erzielen.

Die Gemeindepräsidentin, eine freundliche aber entschlossene Frau, beauftragt darum den Bausekretär, sich zusammen mit dem technischen Leiter der Ilishornbahnen um die Sache zu kümmern. Als Fachmann zieht sie den Informatiker ihrer Kollegin bei.

2.1.2 Erste Auslegeordnung

Beim ersten Gespräch der Arbeitsgruppe purzelten die verschiedenen Stichworte und Argumente anfänglich noch wild durcheinander: Mit welchem Programmpaket arbeitet der nationale Verband? Meint man nur die Bahnunternehmung oder jede einzelne Bahn, jeden Skilift? Die einzelnen Bahnen, aber auch die Gebäude sind ja bereits in der amtlichen Vermessung enthalten. Wie kann man diese Daten brauchen? Was passiert, wenn Daten

ändern, wenn neue Daten hinzukommen? Aber mein Programmpaket versteht nur DXF! Und ich nur Bahnhof!

Das gab das Zeichen, um kurz inne zu halten. Man erinnerte sich an die alte römische Devise «Divide et impera» und begann Ordnung zu machen, um der Sache Herr zu werden. Im Vordergrund standen zuerst folgende Fragen:

- ◆ Wer braucht welche Daten?
- ◆ Wer erfasst sie und führt sie nach?

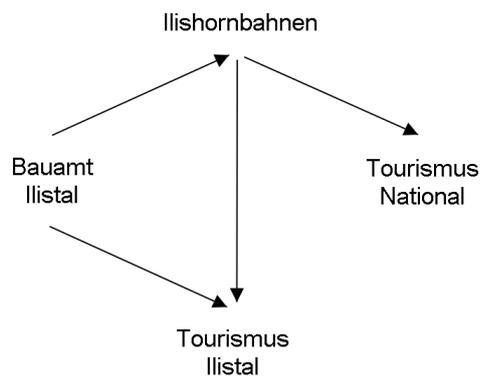


Abbildung 1: Die verschiedenen Beteiligten und die Datenströme.

Aber wie kommen die Daten nun vom Bearbeiter zum Nutzer? E-Mail, FTP, DXF, ASCII – schon ging's wieder los. Der Informatiker empfahl, diese Frage mal etwas zurückzustellen und sich darum zu kümmern, wie die Daten modelliert seien. Modelliert? Wir wollen doch eine Informatiklösung und kein Holzmodell für unser schönes Ilistal. Die Frage, was ein Datenmodell ist, sprengte den Rahmen der ersten Sitzung. Nur soviel: Es soll beschreiben, wie die Daten aufgebaut sind. Welche Eigenschaften haben die verschiedenen Objekte? Welche Objekte stehen mit welchen anderen Objekten in Beziehung? Und das nicht einfach als Aufsatz, sondern in einer klaren, präzisen bildlichen oder formalen Sprache!

2.2 Erste Gehversuche

2.2.1 Der nationale Tourismusverband hat Vorarbeit geleistet

Der nationale Tourismusverband bietet mit seinem Programmpaket NatTourSys einen Überblick zu den verschiedenen Billetten der Bergbahnen an. Die Billette werden durch die einzelnen Bahngesellschaften herausgegeben. Der Tourist interessiert sich aber vor allem dafür, auf welchen Bergbahnen welche Billette gültig sind. Bevor sie in ihr eigenes Projekt einsteigen, wollen sich die Ilistaler zunächst eine Übersicht verschaffen.

Eines ist klar: Im Zusammenhang mit dem Programmpaket muss man über Bergbahnen, Bahngesellschaften und Billette sprechen.

Was meint man wirklich, wenn man von Billetten spricht? Wirklich das einzelne Billett, das verkauft wird? In dieser Anwendung sicher nicht. Man möchte die einzelnen Arten von Billetten beschreiben. Besser sprechen wir darum gleich von Billettarten. Die verschiedenen

Gegenstände haben offensichtlich weitergehende Eigenschaften: Bei den Billettarten kann man z.B. von einem Preis und von einer Gültigkeitsdauer sprechen.

- ◆ **Bergbahn** – Eine Bergbahn transportiert Passagiere zwischen ihrer Tal- und ihrer Bergstation. Ein Beispiel für eine Bergbahn ist die Standseilbahn Ilistal – Ilishorn. Es gibt aber auch Zahnrad-, Luftseil- und Gondelbahnen sowie Ski- und Sessellifte. Auch der neuartige Schneebus kann als Bergbahn angesehen werden. Jede Bergbahn besitzt einen Namen.
- ◆ **Billettart** – Eine Billettart ist eine bestimmte Art eines Billetts. Beispiele für Billettarten sind der Sportpass zu 195 Talern, der sieben Tage lang auf allen Bahnen in der ganzen Region Ilistal gültig ist, oder der am Ausstellungstag gültige «Dino-Pass» für den Ponylift zu 10 Talern.
- ◆ **Bahngesellschaft** – Eine Bahngesellschaft betreibt Bergbahnen. Sie besitzt einen Namen und manchmal auch eine Kurzbezeichnung. Ein Beispiel sind die Ilishornbahnen mit der Abkürzung IhB. Jede Bahngesellschaft erhält einen bestimmten Anteil an den Verkäufen jener Billette, die auf ihren Bahnen gültig sind. Eine Bahngesellschaft kann die Tochter- oder Muttergesellschaft einer anderen sein.
- ▶ Der **Objektkatalog** einer Anwendung listet die Gegenstände auf, die dabei wesentlich sind, und beschreibt sie in Worten möglichst gut.

Beschreibt man nun alle Eigenschaften der Gegenstände in Worten, wird bald einmal die Übersicht relativ schwierig. «Ein Bild sagt mehr als tausend Worte» kommt einem in den Sinn. Ein Objektdiagramm – das wär's! Aber eigentlich wollen wir ja gar nicht die einzelnen Objekte beschreiben. Wir möchten vor allem aufzeigen, welche gleichartigen Gegenstände es gibt und welche Eigenschaften sie haben.

Mit einem solchen Diagramm sieht man das Wichtigste ganz gut:

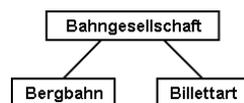


Abbildung 2: Ein erster Ansatz eines Datenmodells.

- ▶ Bergbahn, Billettart und Bahngesellschaft sind **Objektklassen** (Kästchen). Zwischen ihnen bestehen **Beziehungen** (Verbindungslinien). Die Gesamtheit der Definitionen für die Klassen und ihre Beziehungen bildet das **Datenmodell**. Ihre bildliche Darstellung erfolgt mit **Klassendiagrammen**.

Mit Objektklassen verwandte Begriffe sind: Entitätsmenge, Tabelle, Typ, ...

Mit Beziehungen verwandte Begriffe sind: Assoziation, Verweis, Verknüpfung, (gegenseitige) Zeiger, ...

Mit Datenmodell verwandte Begriffe sind: (konzeptuelles) Schema, Datenbeschreibung, ...

Objektklassen werden mit Substantiven bezeichnet. Es wird die Einzahl verwendet, weil man ausdrücken möchte, dass jedes einzelne Objekt (z.B. also jede einzelne Bergbahn) die mit der Klasse beschriebenen Eigenschaften hat.

- ▶ Jede einzelne Bergbahn, jede Bahngesellschaft, jede Billettart wird durch ein konkretes **Objekt** beschrieben. Die Objekte sind die **Daten**, deren Struktur und Zusammenhänge durch das Modell beschrieben werden.

Mit Objekt verwandte Begriffe sind: Exemplar, Instanz, Ausprägung, Datensatz, Zeile, Tupel, Eintrag, ...

Jede Bergbahn wird von einer Bahngesellschaft betrieben. Diese bietet bestimmte Billettarten an. Ohne zusätzliche Informationen müssen sie jeweils für alle ihre Bergbahnen gültig sein. Damit wird man kaum zufrieden sein, da grössere Gesellschaften durchaus Billettarten herausgeben, die nur für einen Teil ihrer Bahnen gelten. Als nahe liegende Idee führt man eine weitere Beziehung zwischen Bergbahn und Billettart ein. Es wird also bei jeder Billettart aufgezählt, für welche Bergbahnen sie gilt:

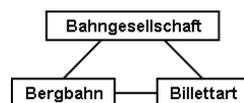


Abbildung 3: Das Datenmodell wurde um eine Beziehung Bergbahn – Billettart erweitert.

Häufig sind aber mehrere Billettarten (z.B. Tageskarte, Wochenkarte, etc.) im gleichen Bereich gültig. Mit dem bis jetzt formulierten Modell müssten die Zuordnungen für jede Billettart einzeln erstellt werden. Das ist eher mühsam und fehleranfällig. Darum hat wohl auch der nationale Tourismusverband ein etwas verfeinertes Modell gewählt:

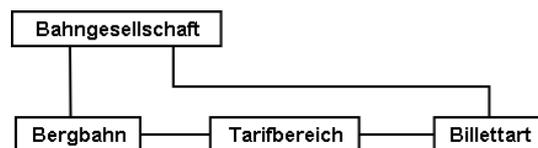


Abbildung 4: Revidiertes Datenmodell. Der Knick in der Verbindungslinie zwischen Bahngesellschaft und Billettart hat nichts zu bedeuten.



Es lohnt sich, zunächst zu überlegen, welche Objektklassen für die Problemstellung nötig sind und wie sie zueinander in Beziehung stehen. Dabei sind die Eigenschaften der Objekte noch relativ unbedeutend. Wichtiger ist, dass geeignete Begriffe gesucht werden.

2.2.2 Wie viele Bahnen betreibt eine Bahngesellschaft?

Mehrere Bergbahnen können einer Bahngesellschaft zugeordnet werden. Einer Bahngesellschaft können umgekehrt mehrere Bergbahnen zugeordnet werden. Mehrere? Wie viele genau?

- Die **Kardinalität** hält fest, wie viele Objekte der andern Art einem Objekt der einen Art zugeordnet sind.

In der Grafik wird die minimale und maximale Zahl der zulässigen anderen Objekte am Ende der Beziehungslinie bei der Klasse der anderen Objekte angemerkt. Ist die Zahl nach oben nicht beschränkt ist, schreibt man einen Stern (*) oder lässt die Angabe weg.

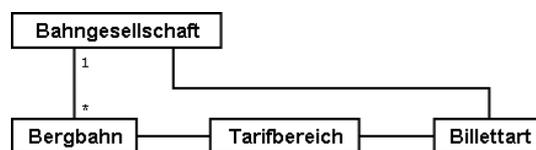


Abbildung 5: Eine Bergbahn wird von genau einer (1) Bahngesellschaft betrieben. Umgekehrt kann aber eine Bahngesellschaft beliebig viele (*) Bergbahnen betreiben.

2.2.3 Bergbahnen, Bahngesellschaften und Abonnements haben Eigenschaften

Selbstverständlich ist es für die geplante Anwendung nötig, dass eine Bergbahn, eine Bahngesellschaft, usw. detaillierter beschrieben wird. Eine Gesellschaft wird einen Namen und (bei Bahnen typisch) eine Kurzbezeichnung haben (z.B. Ilishornbahnen, IhB).



Abbildung 6: Die Objektklasse Bahngesellschaft mit Namen und Kurzbezeichnung.

- Name und Kurzbezeichnung bezeichnen **Attribute** der Objektklasse Bahngesellschaft.

Mit Attribut verwandte Begriffe sind: Kolonne, Feld, Eigenschaft, ...

Bei diesen beiden Attributen ist es bereits durch die Bezeichnung recht offensichtlich, von welcher Art sie sind: Texte. Beim Preis einer Billettart werden weitere Angaben schon etwas wichtiger: Franken, Euro, Dollar, Ahländer Taler? Bei der Gültigkeitsdauer würde es vor allem dann anspruchsvoller, wenn sie nicht einfach mit einer Anzahl von Tagen beschrieben werden kann. Will man bei einer Bahn die Länge angeben, ist es natürlich auch von Belang, ob sie in Meter oder Kilometer beschrieben wird. Für die bearbeitenden Programme ist es wichtig zu wissen, wie lang die vorgesehenen Textattribute werden dürfen oder in welchem Bereich die vorgesehenen Zahlen liegen dürfen.

- Der **Typ** eines Attributs beschreibt, welche Werte das Attribut annehmen kann und welche Bedeutung sie haben.

Ein mit Typ verwandter Begriff ist Wertebereich.

Objektklasse Bahngesellschaft

Name: Text
Länge: Höchstens hundert Zeichen

Kurzbezeichnung: Text
Länge: Höchstens zehn Zeichen

Abbildung 7: Die Objektklasse «Bahngesellschaft» besitzt einen Namen und eine Kurzbezeichnung. Der Typ der Eigenschaft «Name» ist ein Text mit maximal hundert Zeichen. Für die Eigenschaft «Kurzbezeichnung» sind dagegen höchstens zehn Zeichen zugelassen.

Man kann sich aber auch gut andere Attributtypen vorstellen:

Objektklasse Billettart

Name: Text mit höchstens hundert Zeichen

Preis: Zahl
Genauigkeit: Zwei Stellen nach dem Komma
Zulässiger Bereich: Zwischen 0 und 5000
Einheit: Ahländer Taler

Abbildung 8: Die Objektklasse Billettart mit ihren Eigenschaften und deren Typen.

Anders als eine Billettart oder eine Bergbahngesellschaft ist die Talstation einer Bergbahn ein Gegenstand, der real an einem bestimmten Ort existiert. Orte werden sinnvollerweise mittels Koordinaten innerhalb eines bestimmten Koordinatensystems, z.B. des Landessystems, beschrieben.

Objektklasse Bergbahn

Name: Text mit höchstens hundert Zeichen

Lage der Talstation: Punkt
Koordinatensystem: Ahländer Landeskoordinaten

Lage der Bergstation: Punkt
Koordinatensystem: Ahländer Landeskoordinaten

Abbildung 9: Die Objektklasse Bergbahn mit ihren Eigenschaften und deren Typen.

Für jede Eigenschaft wird somit ein passender Attributtyp festgelegt. Im Fall einer Skipiste ist der Schwierigkeitsgrad eine Aufzählung. Der Verlauf der Skipiste ist dagegen eine gerichtete Linie in Ahländer Landeskoordinaten. Details zu verschiedenen Typen werden in Kapitel 6 besprochen.

Objektklasse Skipiste

Verlauf:	Gerichtete Linie Koordinatensystem: Ahländer Landeskoordinaten
Schwierigkeitsgrad:	Aufzählung Mögliche Werte: blau, rot, schwarz

Abbildung 10: Die Objektklasse Skipiste mit ihren Eigenschaften und deren Typen.**2.2.4 Modelle? Ilistal will Daten!**

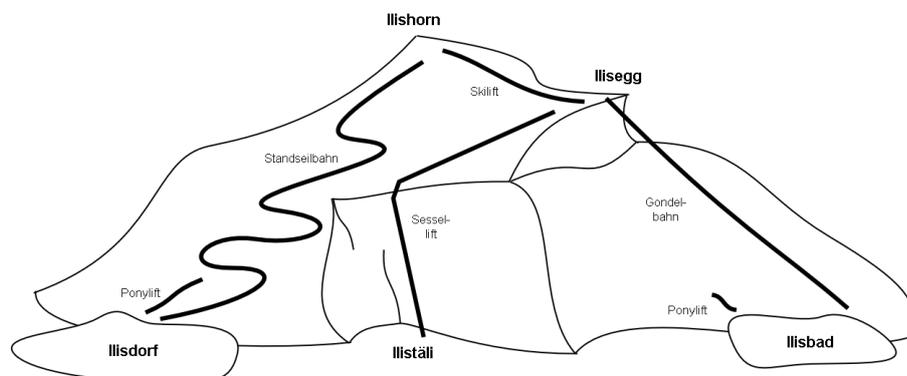
Nach all diesen doch eher theoretischen Dingen drängen die Ilistaler nun auf Taten. Eine Anfrage beim nationalen Tourismusverband ergab, dass dieser ein einfaches Programm zur Datenerfassung gemäss seinen Anforderungen zur Verfügung stellt. Mit diesem können die Daten im INTERLIS-Format exportiert und dann dem nationalen Tourismusverband geschickt werden. Der Informatiker wendete zwar ein, dass damit höchstens ein erster Test gemacht werden könne und man die Daten nachher im Programmpaket der Ilishornbahnen oder in demjenigen des Bauamtes halten sollte. Diesen Test wollte man aber schon machen. Schliesslich sollte damit ja nicht soviel Arbeit verbunden sein. Denn so umfangreich sind die Ilishornbahnen auch wieder nicht, und die Anzahl Billettarten hält sich auch in Grenzen.



Hopp-Hopp-Aktionen machen nur dann einen Sinn, wenn sie wirklich nicht mit viel Arbeit verbunden sind.

Zu den Ilishornbahnen gehören folgende Bergbahnen:

- ◆ Standseilbahn Ilistdorf – Ilishorn;
- ◆ Gondelbahn Ilistbad – Ilistegg;
- ◆ Skilift Ilistegg – Ilishorn;
- ◆ Sessellift Ilisttäli – Ilistegg;
- ◆ Ponylifte in Ilistdorf und Ilistbad.

**Abbildung 11:** Die Ilishornbahnen betreiben mehrere Bahnen.

Die Ilishornbahnen bieten dafür folgende Billettarten an:

- ◆ Einzelbillette für die Standseilbahn (Preis für eine einfache Fahrt: 10 Taler; für eine Retourfahrt: 18 Taler);
- ◆ Einzelbillette für die Gondelbahn (Preis für eine einfache Fahrt: 8 Taler; für eine Retourfahrt: 14 Taler);
- ◆ den Wanderer-Pass für die Standseilbahn und die Gondelbahn (Preis für einen Tag 15 Taler; für sieben Tage 55 Taler);
- ◆ den Sportpass für alle Bahnen (Preis für einen Tag: 40 Taler, für zwei Tage: 70 Taler, für sieben Tage: 195 Taler; für ein ganzes Jahr: 635 Taler);
- ◆ die «Dino-Tageskarte» (10 Taler) und den «Wochenpass Ilosaurus Maximus» (45 Taler) für die Ponylifte.

2.2.5 Ilistal sendet

Für den Test konnte mit dem Programm eine Datei erstellt werden, die alle Daten enthält.

- ▶ Die einfachste Transferart ist der **Volltransfer**, bei dem alle Daten vollständig übermittelt werden.

Ein kurzer Blick in die Datei zeigte zwar viel Unverständliches, aber immerhin fand man die Texte «Ilishornbahnen», gleich daneben «IhB», und auch der Abonnementspreis war leicht zu finden.

Gleich noch ein Test: Der Jahrespreis für den Sportpass wird von 635 auf 600 Taler gesenkt und mit der Funktion Nachlieferung eine neue Datei erstellt. Der Anfang sieht zwar noch gleich aus. Die Texte «Ilishornbahnen» und «IhB» fehlen. Aber hier, fast am Schluss – das könnte der neue Preis sein!

- ▶ Dank der **inkrementellen Nachlieferung** müssen nach einer Änderung nicht alle Daten, sondern nur die geänderten Objekte übermittelt werden.

Beide Dateien wurden dann wie vereinbart dem Tourismusverband geschickt. Dieser konnte sie offenbar problemlos lesen. Einwand des Informatikers: Das ist ja auch nicht verwunderlich. Solange wir genau die Daten erfassen, die der Verband will, und dies erst noch mit einem Programm, das der Verband zur Verfügung stellt, kann man das ja schon erwarten. Aber wir Ilistaler wollen doch mehr! Und wir möchten doch wenn immer möglich unsere bisherigen Programmpakete einsetzen.

2.3 Ilistal will mehr

2.3.1 Zielsetzung

Ilistal möchte also nicht einfach den gleichen Service anbieten, wie dies der nationale Tourismusverband tut. Folgende Zusatzleistungen stehen im Vordergrund:

- ◆ Anzeige der aktuellen Betriebs- und Wartezeiten der verschiedenen Bahnen und ob sie von Wanderern und Schlittlern benützt werden können;
- ◆ Anzeige der Pisten inkl. Schwierigkeitsgrad und Befahrbarkeit;
- ◆ bildliche Darstellung (inkl. Anzeige der Wälder und Strassen);
- ◆ Anzeige der Gasthäuser der Region;
- ◆ Anzeige, wo sich die Gebäude mit welcher Postadresse befinden.

2.3.2 Ilistal nutzt Vorhandenes

Die für die bildliche Darstellung nötigen Daten der Wälder und Strassen möchte man natürlich nicht selbst erfassen, denn das Bauamt hat schliesslich die Daten der amtlichen Vermessung, welche auch Wälder und Strassen enthalten. Und das Bauamt hat doch damit begonnen, die Gebäudeadressen gemäss der neuen Norm zu erfassen. Nun macht es wenig Sinn, im Datenmodell von Ilistal all diese Definitionen zu wiederholen. Man möchte einfach die vorhandenen Modelle der amtlichen Vermessung und der Gebäudeadressen nutzen.

- ▶ Ein Datenmodell ist nicht eine isolierte Beschreibung, sondern kann auf bereits bestehenden Datenmodellen aufbauen.

Im Sinne des Aufbaus verwandte Begriffe zu Datenmodell sind: Module, Pakete, Packages, ...

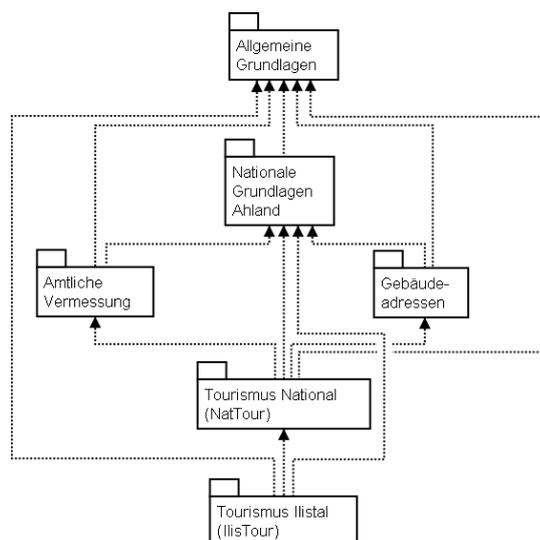


Abbildung 12: Das Ilistaler Tourismus-Datenmodell (IlistTour) braucht nicht alles selbst zu definieren. Stattdessen baut es auf anderen Modellen auf: Es verwendet Teile des nationalen Tourismus-Modells (NatTour), der nationalen Grundlagen von Ahland, der amtlichen Vermessung, der Gebäudeadressen sowie allgemeine Grundlagen. Die gestrichelte Linie mit gefülltem Pfeil steht für die Abhängigkeit. Häufig wird die allgemeine Grundlage wie hier oben, der Spezialfall unten gezeichnet. Die umgekehrte Schreibweise ist jedoch ebenfalls verbreitet.

2.3.3 Ilistal geht weiter als der nationale Verband

Das Modell des nationalen Tourismusverbandes wollen die Ilistaler aber nicht einfach so nutzen, wie es ist. Damit man eine bildliche Darstellung machen kann, muss doch bei jeder Bergbahn auch der Trasseeverlauf beschrieben sein. Zudem möchte man festhalten, ob die Bahn durch Wanderer und Schlittler benutzbar ist, wann sie in Betrieb ist und wie lange die aktuelle Wartezeit ist. Es wäre nahe liegend, eine eigene Klasse für die Ilistaler Bergbahnen zu definieren. Sollen dabei die Attribute der Klasse Bergbahn des nationalen Verbandes wiederholt werden? Und da gibt es noch die Beziehung zwischen Bergbahnen und Tarifbereichen. Was würde eine eigene Klasse für diese Beziehung heissen?

Zum Glück gibt es für solche Fälle die Vererbung.

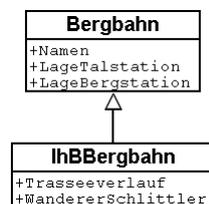


Abbildung 13: Eine IhBBergbahn ist eine spezielle Bergbahn mit zusätzlichen Attributen: Trasseeverlauf sowie Benutzbarkeit für Wanderer und Schlittler. Die ausgezogene Linie mit weisser Pfeilspitze bezeichnet die Spezialisierung.

- Die Klasse der Ilistaler IhB-Bergbahn ist eine **Erweiterung** der Klasse der Bergbahnen. Sie **erbt** damit alle Eigenschaften der Bergbahnen und fügt weitere hinzu. [Details der Vererbung werden in Kapitel 5 beschrieben].

Mit Erweiterung verwandte Begriffe sind: Spezialisierung, Subklasse, ...

Wäre es nun richtig, in der Klasse IhBBergbahn gleich auch die Attribute Betriebszeit und aktuelle Wartezeit aufzunehmen? Wäre die Betriebszeit direkt ein Attribut von IhBBergbahn, könnte für jede Bahn genau eine, typischerweise die aktuelle Betriebszeit festgehalten werden. Der Betriebsleiter legt aber die Betriebszeiten jeweils zu Saisonbeginn fest: In der Vorsaison sind einzelne Lifte noch nicht in Betrieb, andere haben Mittagspause; über die Weihnachtstage ist voller Betrieb von 9 bis 15.30 Uhr; ab Mitte Februar – wenn die Tage länger werden – wird die Betriebszeit schrittweise bis 16.30 ausgedehnt. Je nach Schnee- und Wetterverhältnissen wird dann der Betrieb einzelner Bahnen vorübergehend stillgelegt.

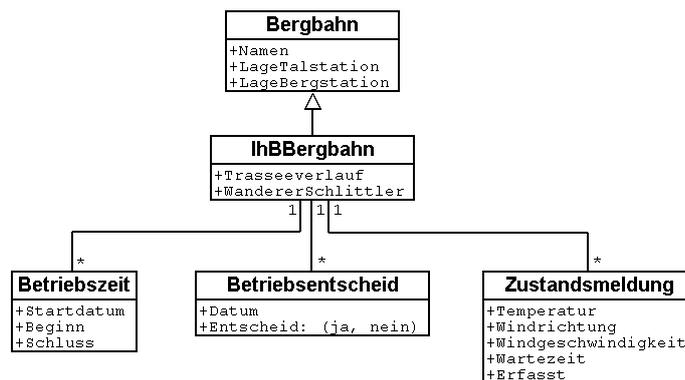


Abbildung 14: Betriebszeiten sind neu als eigenständige Objekte definiert.

Legt man zudem noch fest, dass eine bestimmte Betriebszeit für verschiedene Bahnen gelten kann, kann der Erfassungsaufwand noch weiter gesenkt werden. Bei den Wartezeiten macht dies natürlich keinen Sinn. Eine zu einem bestimmten Zeitpunkt beobachtete Wartezeit muss derjenigen Bahn zugeordnet werden, bei der entsprechend gewartet werden muss. Warum hält man die Wartezeit also nicht direkt auf der IhBBergbahn fest? Folgende Gründe sprechen dagegen:

- ◆ Dank der Speicherung der Wartezeiten als eigenständige Objekte können sie zu einem späteren Zeitpunkt (z.B. für Statistiken) ausgewertet werden.
- ◆ Der Änderungsrhythmus und die Zuständigkeit für die Werte sind ganz anders als bei den Attributen der IhBBergbahn-Klasse.



Bei den Eigenschaften, die auf den ersten Blick einer Klasse zugeordnet werden, muss immer genau überlegt werden, ob dies wirklich korrekt ist, oder ob sie besser in selbständige, über Beziehungen zugeordnete Klassen verschoben werden.

Bei diesen Überlegungen ist der tatsächliche Sachverhalt und nicht die Nutzung z.B. für Darstellungen massgebend. Wichtig sind aber auch die organisatorischen Verhältnisse. Wer ist für die Nachführung der Daten zuständig? In welchem Rhythmus werden sie nachgeführt?

Im Modell des nationalen Verbandes sind die einzelnen Bergbahngesellschaften für die Nachführung ihrer Daten verantwortlich. Das Ilistaler Modell möchte – was die Bergbahnen betrifft – das Modell des nationalen Verbandes zwar nutzen, es aber für die Ilishorn-Bergbahnen erweitern.

- ▶ Datenmodelle werden in **Themen** gegliedert, um den organisatorischen Verhältnissen (z.B. unterschiedliche Verantwortlichkeiten und Nachführungsrhythmen) gerecht zu werden.

Das Ilistaler Modell erweitert darum das vom nationalen Verband vorgegebene Thema Bergbahnen zu IhBBergbahnen. In dieser lokalen Erweiterung ist definiert, dass die Klasse IhBBergbahn die Klasse Bergbahn spezialisiert und um zusätzliche Attribute erweitert.

Da die Betriebszeiten, die Betriebsentscheide und die Zustandsmeldungen teilweise durch verschiedene Stellen, vor allem aber in völlig anderem Rhythmus erfasst werden, werden sie je in eigenen Themen (IhBPlanung, IhBBetrieb, IhBAktuell) definiert.

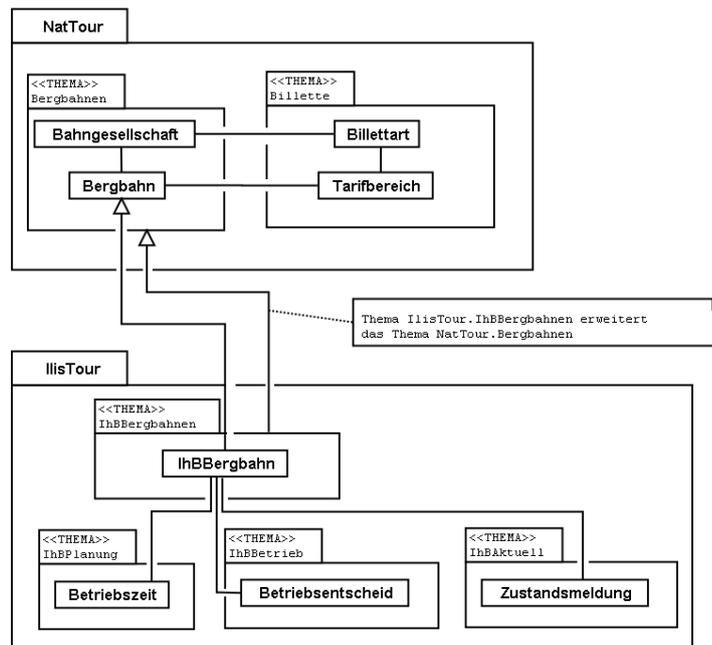


Abbildung 15: Das Ilistaler Modell (IlistTour) erweitert das Modell des nationalen Tourismus-Verbands (NatTour). IlistTour erbt das Thema Bergbahnen von NatTour, erweitert die Klasse Bergbahn zu IhBBergbahn und fügt weitere Themen für Planung, Betrieb und Aktuelles an.

- Vererbung gibt es nicht nur im Kleinen (Objektklassen), sondern auch im Grossen (ganze Themen).

2.3.4 Ilistaler Spezialitäten

Zusätzlich möchten die Ilistaler auch Pisten und Gasthäuser beschreiben. Sie fügen dem Ilistaler Modell darum noch weitere Themen zu.

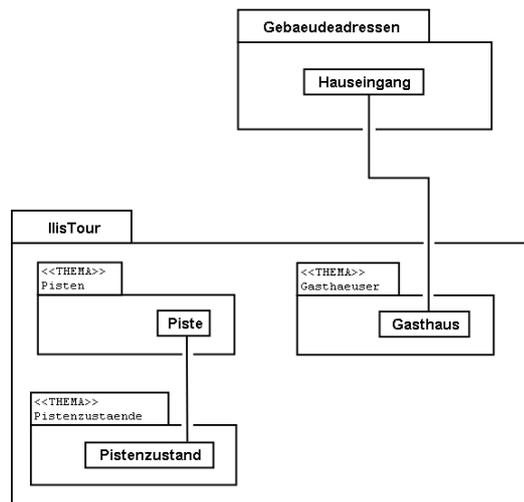


Abbildung 16: Das Ilistaler Tourismus-Modell wird um weitere Themen ergänzt.

Besonders bei den Gasthäusern stellen sich wieder Fragen. Wie soll z.B. der Schnellimbiss INTERLUNCH bildlich dargestellt werden? Man weiss zwar, dass er an der Dorfstrasse 27 liegt. Aber damit kann man kein Symbol auf dem Plan platzieren! Die Lösung liegt in der Nutzung der Gebäudeadressen. Dort gibt es eine Klasse Hauseingang, die auch ein Lageattribut (in Landeskoordinaten) aufweist. In der Klasse Gasthaus wird darum gar keine Adresse geführt, sondern eine Beziehung zum Hauseingang definiert. Konkret wird das Objekt, welches dem Hotel Sonne entspricht, mit dem Hauseingang-Objekt verbunden, das die Dorfstrasse 27 beschreibt.

2.3.5 Wie implementieren die Ilistaler ihre Spezialitäten?

Ein Konzept regelt die Anforderungen, nicht aber die Implementation. In der Implementation ist man grundsätzlich frei. Die Ilishornbahnen haben sich für ein standardisiertes Programmpaket (LiftSys) entschieden, das allerdings nur Daten gemäss dem erweiterten Modell behandeln kann. Es ist aber ohne weiteres zulässig, auf die Klasse Bergbahn zu verzichten und deren Attribute gleich in der Klasse lhBBergbahn einzufügen.

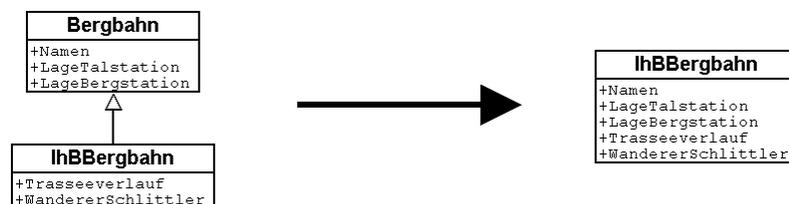


Abbildung 17: Das Programmpaket für den Ilistaler Tourismus braucht sich nur grob ans konzeptuelle Modell zu halten. Es kann zum Beispiel intern zwei Objektklassen zu einer einzigen zusammenfassen. Wichtig ist nur, dass das Paket in der Lage ist, die Daten in jenem Format zu liefern, das dem konzeptuellen Modell entspricht.

Analog zur Behandlung der Klassen gemäss Konzept stellen sich verschiedene weitere Fragen, wie ein bestimmtes Computersystem die Vorstellungen umsetzt, die mit dem konzeptuellen Modell verbunden sind.

2.3.6 Wie schicken die Ilistaler ihre Daten an den nationalen Tourismusverband?

Nachdem das LiftSys-Programmpaket eingerichtet und die Daten erfasst sind, stellt sich wieder die Frage, wie die Daten dem nationalen Verband übermittelt werden können. Dieser will natürlich nicht alle Daten, sondern nur diejenigen, die ihn interessieren. Der nationale Verband ist z.B. weder an Pisten noch an der Eignung für Wanderer und Schlittler interessiert.

- ▶ Ein INTERLIS-Datentransfer umfasst immer die Daten von einem oder mehreren Themen.

Die Ilistaler wollen darum dem nationalen Verband die Daten der Themen Bergbahnen und Billette schicken. Aber wie kann ein Programmpaket eine korrekte Transferdatei erstellen – der Hersteller kannte doch die Spezifikationen des Tourismusverbandes gar nicht? Die Lösung liegt im *modellbasierten Transfer*.

- ▶ Bei einem **modellbasierten Transfer** gibt es nicht ein ganz bestimmtes **Transferformat**. Vielmehr richtet sich das Format nach dem Datenmodell.

Jede Modellierungsmethode (z.B. INTERLIS, oder die Definitionen, mit denen ein bestimmtes Programmpaket eingerichtet wird) stellt bestimmte Ausdrucksmittel (Objektklassen, Attribute, Typen, Beziehungen, Tabellen, Kolonnen, usw.) zur Verfügung. Für jedes solche Ausdrucksmittel wird unabhängig vom konkreten Datenmodell geregelt, welche Wirkungen es auf den Transfer hat. Von einem konkreten Transferformat, also der genauen Reihenfolge der Zeichen, welche die jeweiligen Daten repräsentieren, kann man somit erst sprechen, wenn das zugehörige Datenmodell bekannt ist. Ja, das Transferformat ergibt sich direkt aus dem Datenmodell.

Wäre LiftSys in der Lage, das interne Datenmodell direkt gemäss dem konzeptuellen Datenmodell aufzubauen, und würde es zusätzlich die Umsetzung der Daten in Transferdateien gemäss den Spezifikationen von INTERLIS unterstützen, wäre alles kein Problem. Die Transferdateien könnten genau so einfach erstellt werden wie mit dem Testprogramm des Verbandes.

Das Programmpaket des Bauamtes (BauSys) unterstützt zum Beispiel die Erstellung von INTERLIS 2-konformen Dateien. Es kennt aber nur einzelne Tabellen, die jeweils mehrere Kolonnen aufweisen können. Da die Formatierungsregeln von INTERLIS so aufgebaut sind, dass die Vererbungsstruktur sich in der Transferdatei nicht direkt spiegelt, könnten mit BauSys direkt korrekte Dateien erstellt werden. Die Umsetzung von den internen in die externen Daten kann man sich wie folgt vorstellen:

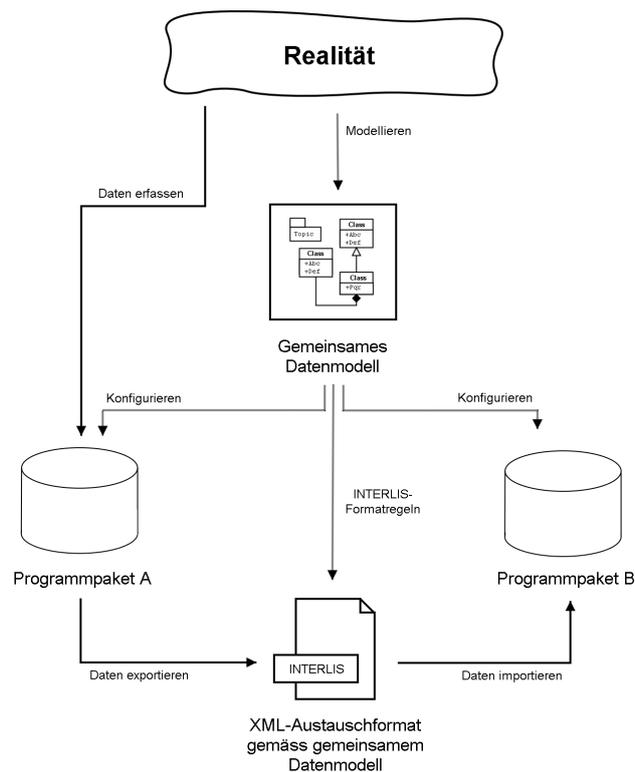


Abbildung 18: Die internen Daten des Programmpaketes A werden in eine Transferdatei umgesetzt, deren Aufbau gemäss den Formatregeln von INTERLIS aus dem Datenmodell folgt. Die Daten können danach in Programmpaket B importiert werden. Voraussetzung ist, dass die beteiligten Programmpakete entsprechend dem Datenmodell konfiguriert worden sind.

Das LiftSys unterstützt aber INTERLIS nicht. Was nun? Müssen sich die Ilishornbahnen mit dem Kauf eines neuen Programmpaketes beschäftigen? Der Ausweg ist offensichtlich: LiftSys exportiert die Daten in einem anderen Format, diese werden dann mit einem Konversionsprogramm nach INTERLIS umgeformt. Das Konversionsprogramm kann entweder spezifisch für das konkrete Datenmodell oder allgemein als modellbasiertes Werkzeug realisiert sein.

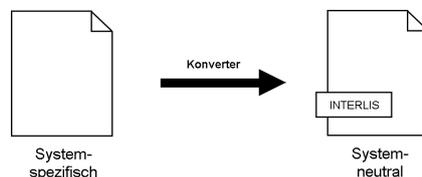


Abbildung 19: Ein Konverter erstellt INTERLIS-Dateien aus einem Format, das spezifisch für ein bestimmtes Computersystem ist.

Nachdem alles glücklich funktionierte, wird die Datei an den nationalen Verband geschickt. Das Echo: «Fast gut – beim Namen des Sesselliftes auf die Ilistal gibt es aber ein Problem!» Uff – das kennen wir doch auch aus verschiedenen E-Mails: «Ilistäli»; immer diese Umlaute.

Zwei Dinge sollten klar unterschieden werden:

- ▶ Der **Zeichenvorrat** legt fest, welche Zeichen bei Text-Attributen überhaupt verwendet werden dürfen.

- ▶ Die **Zeichencodierung** legt fest, welches Bitmuster das Zeichen im Computer repräsentiert.

Die Umlaute gehören zum erlaubten Zeichenvorrat von INTERLIS. Aber bei der Konversion wurde vergessen, die Zeichencodierung der Daten, die vom LiftSys kamen, korrekt anzugeben. Nach der Korrektur erhält Ilistal ein positives Echo vom Verband.

2.3.7 Was macht der nationale Tourismusverband mit den Ilistaler Daten?

Über etwas sind nun die Ilistaler ein wenig erstaunt: Was hat das Computersystem des nationalen Tourismusverbands (NatTourSys) wohl mit den zusätzlichen Attributen angefangen – etwa mit der Eignung für Wanderer und Schlittler oder gar dem Trasseeverlauf? Die Lösung klingt simpel: NatTourSys hat sie einfach ignoriert.

- ▶ **Polymorphes Lesen** erlaubt, dass die Daten gemäss einem «reduzierten» Modell, d.h. einem Modell, das zusätzliche Erweiterungen noch nicht kennt, gelesen werden können.

Die Ilistaler haben die Daten zwar so geschickt, dass sie alle Erweiterungen gemäss dem Ilistaler Modell enthalten. Die Transferregeln von INTERLIS sorgen dafür, dass die Daten dennoch gemäss dem Modell des nationalen Tourismusverbandes gelesen werden können, ohne dass das Leseprogramm wegen der zusätzlichen Daten aus dem Takt gerät. Bedingung ist nur, dass das Modell, gemäss dem die Daten erstellt wurden, eine Erweiterung jenes Modells ist, das der Empfänger der Daten benutzt. Das Ilistaler Modell muss also das Modell des nationalen Tourismusverbandes erweitern.

Kapitel 5 erläutert näher, wofür Erweiterungen nützlich sind. Kapitel 8 befasst sich mit den Details des Datentransfers.

Dabei ist es auf der lesenden Seite möglich, dass die Daten direkt mit dem Programmpaket des Empfängers gelesen werden oder dass auch hier ein Konversionsprogramm zwischengeschaltet wird. Und auch hier gilt wieder, dass die konkreten Zeichen der Text-Attribute korrekt umgesetzt werden. Das «ä» von Ilistäli kann durchaus im LiftSys, auf der Transferdatei und im NatTourSys je unterschiedlich codiert sein. Für die jeweiligen Programme ist es immer klar, dass es ein «ä» ist.

2.4 Ilistal hat es geschafft

2.4.1 Systemübersicht

Beim Internet hat man sich für eine relativ einfache Lösung entschieden: Der Situationsplan wird als statisches Bild durch das Programmpaket LiftSys erstellt und danach einem Web-Präsentationssystem (WebSys) zur Verfügung gestellt. Um den aktuellen Zustand der Bah-

nen abfragen zu können, werden bestimmte Bereiche im Bild markiert. Wird mit der Maus auf einen dieser Flecken geklickt, erscheinen die aktuellen Zustandsdaten der betreffenden Bahn. Zudem sollen die Hotels mit freien Zimmern speziell markiert werden.

2.4.2 Für die Web-Seite ist nur der aktuelle Zustand interessant

Die Ilistaler haben sich Mühe gegeben und ihr Modell vor allem auch für die betrieblichen Daten der Bahnen und Pisten sauber strukturiert. Nur ist leider das Programm, welches laufend die Internetseite aktualisiert, nicht in der Lage, aus der Vielzahl von Betriebszeiten, Betriebsentscheiden und Zustandsmeldungen den jeweils aktuellen Zustand abzuleiten. Der Betreiber möchte einerseits die Daten gemäss dem Thema lhBBillette immer dann erhalten, wenn etwas geändert hat. Andererseits möchte er aber über den betrieblichen Zustand der Bergbahnen alle 20 Minuten eine neue Mitteilung erhalten.

- Eine **Sicht** definiert Daten, die der Auffassung eines Nutzers entsprechen und darum aus den Originaldaten abgeleitet werden müssen.

Verwandte Begriffe: View, abgeleitete Daten,

Die verlangte Sicht verbindet Betriebszeiten, Betriebsentscheide und Wartezeiten mit derjenigen Bergbahn, der sie gemäss Beziehung zugeordnet sind und filtert so, dass nur der aktuelle Zustand beschrieben wird.

- Von der Anwendung aus gesehen können Sichtobjekte gleich wie Datenobjekte aufgefasst werden. Sichten werden darum auch mittels Sichtklassen beschrieben.

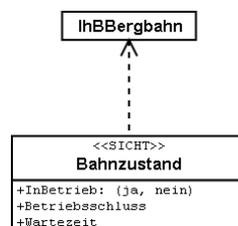


Abbildung 20: Der Bahnzustand ist keine eigenständige Objektklasse, sondern wird über eine Sicht von lhBBergbahn abgeleitet. Die Sicht fasst jene Daten zusammen, die für das Darstellen auf einer Internet-Seite benötigt werden.

2.4.3 Hotels mit freien Betten auf der Web-Seite darstellen

Damit WebSys eine zusätzliche Markierung anbringen kann, in welchen Hotels es noch freie Zimmer gibt, braucht es natürlich die nötigen Informationen. Ähnlich wie bei den Zuständen der Bahnen wird darum auch für Hotels eine Sicht definiert. Sie enthält einerseits die nötigen Daten der Gasthausobjekte, andererseits die Lagekoordinaten gemäss dem zugeordneten Hauseingang.

- Mit INTERLIS können auch die benötigten **Symbole systemneutral definiert** und die Umsetzung von Original- oder Sichtdaten in **Grafik** beschrieben werden.

Leider ist WebSys aber nicht in der Lage, solche Umsetzungsbeschreibungen zu verarbeiten. Es ist aber fähig, die Symboldefinitionen zu lesen. Zudem kann es Daten entgegennehmen, die aussagen, welches Symbol an welcher Stelle dargestellt werden soll und dann die Darstellung vornehmen. Damit kann eine andere Möglichkeit von INTERLIS ausgenutzt werden, die auf LiftSys zur Verfügung steht.

- ▶ Mit INTERLIS können auch bereits umgesetzte Grafikdaten transferiert werden.

LiftSys liefert WebSys darum nicht die Sichtdaten der Hotels, sondern setzt diese selbst in Grafikdaten um. Der genaue Aufbau der Grafikdaten kann wiederum mit Klassen definiert werden. Typische Attribute solcher Grafikdaten sind die Position, der Symbolname, die Farbe.

3. Möglichkeiten zur Beschreibung von Daten

3.1 Beschreibung als Grafik: Unified Modeling Language UML

Ein Diagramm eignet sich gut dafür, sich selber und anderen schnell einen Überblick zu einem Modell zu verschaffen. Daher werden in der Informatik bereits seit Mitte der 1970er Jahre graphische Notationen eingesetzt, um Modelle zu visualisieren. Allerdings konnte man sich lange Zeit nicht auf eine einheitliche Darstellungsweise einigen, so dass dutzende Notationen gleichzeitig im Einsatz waren. Schliesslich begannen sich zu Beginn der 1990er Jahre drei Methoden (Booch, OMT und OOSE) durchzusetzen. Deren Urheber, die so genannten «drei Amigos», konnten sich 1995/96 auf eine einzige Modellierungssprache einigen, die *Unified Modeling Language (UML)*. Seit 1997 wird die UML als Standard von einem Industriekonsortium, der *Object Management Group*, verwaltet und gepflegt.

Die UML ist flexibel und vielseitig einsetzbar. Wer mit ihrer Schreibweise vertraut ist, kann auf einen Blick Modelle aus den unterschiedlichsten Anwendungsbereichen erfassen. Die Sprache eignet sich übrigens nicht nur zum Modellieren von Daten (mittels der in diesem Papier vorgestellten Klassendiagramme). Sie umfasst auch Elemente für weitere Aspekte eines Systems, zum Beispiel die Aufteilung in Komponenten oder Szenarien für die Benutzung.

Ein wichtiger Faktor für den Erfolg der UML ist ihre Anpassungsfähigkeit: Für viele Anwendungen stellt die Sprache eine ausreichende Infrastruktur bereit, es ist aber jederzeit möglich, die Sprache mit klar definierten, ebenfalls standardisierten Erweiterungsmechanismen an eigene Bedürfnisse anzupassen.

Eine visuelle Sprache wie die UML besitzt zwar viele Vorteile, sie ist aber nicht für alle Anwendungen das ideale Mittel. Sobald es nämlich darum geht, die Details eines grossen und komplexen Modells festzulegen, kann eine Zeichnung auch unübersichtlich und verwirrend werden. Auch ist eine graphische Darstellung eher unpraktisch, um zum Beispiel ein Computersystem einzurichten. In manchen Fällen eignet sich daher eine Textform besser: Die Modelle werden in einer genormten Sprache als Text aufgeschrieben und können so von Mensch und Computer gleichermaßen verstanden werden. Natürlich kann das Bild gleichzeitig weiterhin für den groben Überblick verwendet werden. Zurzeit sind Vorschläge für eine rein textuelle Schreibweise der UML in Diskussion, es wurde jedoch bisher noch kein entsprechender Standard verabschiedet.

3.2 Beschreibung als Text: INTERLIS

In der Schweiz wurde schon vor langem erkannt, dass ein modellbasierter Ansatz beträchtliche Vorteile mit sich bringt. Im Rahmen der amtlichen Vermessung wurde daher 1991 die textuelle Modellierungssprache INTERLIS entwickelt. Ende der 1990er Jahre wurde

INTERLIS 2 ausgearbeitet, welches die erste Version erweitert und modernisiert. Ein halbes Dutzend Spezialisten aus Forschung, Verwaltung und Industrie stellte dabei sicher, dass die Sprache den Anforderungen der Praxis gerecht wird. Zudem wurde grosser Wert darauf gelegt, dass einige Besonderheiten gut unterstützt werden, die sich aus der Anwendung ergeben:

- ◆ Datenmodelle können mit INTERLIS präzise und in einem sehr hohen Detaillierungsgrad beschrieben werden. Dadurch kann ein INTERLIS-Modell direkt zum Bestandteil eines Ausschreibungstextes, eines Vertrags oder einer Verordnung werden, und es ist auch möglich, ein Programmpaket anhand eines INTERLIS-Modells zu konfigurieren.
- ◆ INTERLIS umfasst nicht nur eine Sprache zum Modellieren der Daten, sondern auch ein Verfahren, mit dem aus einem Modell das Transferformat abgeleitet wird, welches zum Austausch der modellierten Daten benutzt wird. Dies ist deswegen besonders wichtig, weil geographische Daten sehr aufwändig zu erfassen und entsprechend wertvoll sind. Zudem liegt ihre Lebensdauer im Bereich von Jahrzehnten, ist also deutlich länger als jene von Informatiksystemen, die meist schon nach wenigen Jahren veralten. Die teuer erhobenen Daten können nur geschützt werden, indem sie in einer Form gehalten werden, die unabhängig von einem konkreten Programm ist.
- ◆ Obwohl INTERLIS natürlich auch für Anwendungen benutzt werden kann, die keinerlei geographischen Bezug besitzen, eignet es sich für räumliche Daten besonders gut. Geometrie wird bereits im Kern der Sprache unterstützt. Auch wurden Vorkehrungen getroffen, aufgrund derer die Qualität und Plausibilität von angelieferten Daten automatisch geprüft werden können.
- ◆ INTERLIS-Datenmodelle sind klar strukturiert: Ein Modell ist in Themen gegliedert, die ihrerseits Objektklassen umfassen. Zur Beschreibung einer Objektklasse gehören die Objekteigenschaften sowie Konsistenz- und Plausibilitätsbedingungen. Sowohl auf Modell- als auch auf Themenstufe können allgemein benutzbare Konstrukte (z.B. Einheiten, Wertebereiche, Hilfs-Strukturen, etc.) deklariert werden.
- ◆ Die föderalistische Struktur der Schweiz mit ihren 26 Kantonen und Halbkantonen, die ihrerseits in weitere Verwaltungseinheiten gegliedert sind, stellt besondere Anforderungen an eine Modellierungssprache. INTERLIS 2 erlaubt es, dass die Beteiligten ein gemeinsames Kernmodell gemäss ihren regionalen Bedürfnissen ausbauen und verfeinern können. Dabei bleibt der problemlose Austausch der Daten auch dann gewährleistet, wenn lokal unterschiedliche Modelle im Einsatz sind. Zudem unterstützt INTERLIS, dass Modelle und Daten in verschiedenen Sprachen abgefasst sein können. Damit kann jeder Landesteil seine jeweilige Amtssprache verwenden.

INTERLIS und UML ergänzen sich: UML-Diagramme sind schnell aufgezeichnet und ideal für eine grobe Übersicht. Mit INTERLIS können die Details präzise und dennoch leicht verständlich festgehalten werden, zudem werden Anforderungen der föderalistischen Geodaten-Praxis speziell unterstützt. Es existieren Werkzeuge, die ein UML-Klassendiagramm automatisch in eine textuelle INTERLIS-Beschreibung umwandeln können. Auch der umgekehrte Schritt, das Darstellen eines INTERLIS-Modells in der UML-Schreibweise, kann automatisch erfolgen.

3.3 Standardisierte Datenmodelle

Es macht wenig Sinn, jedes Datenmodell vollständig von Grund auf neu zu erstellen. Stattdessen können vordefinierte Modelle benutzt und verfeinert werden. Auf verschiedenen Stufen wurden und werden Modelle entwickelt, auf denen Anwendungen aufbauen können:

- ◆ **International** – Im Rahmen von internationalen Normierungsbestrebungen wurde unter anderem ein sehr umfassendes Modell für Basis-Geometrietyper (ISO 19107) entwickelt. Es sind Bestrebungen in Gang, einen Kern herauszuschälen, welcher den wichtigsten Anforderungen der Praxis entspricht (ISO 19137). Hierfür werden die räumlichen Konzepte mehrerer verbreiteter Modellierungssprachen analysiert, unter anderem auch jene von INTERLIS. Wer sich Gedanken darüber macht, wie etwa Kegelschnittflächen oder Dreiecksnetze zu modellieren sind, ist beraten, sich in die Normdokumente bzw. die entsprechenden Entwürfe einzulesen. Die ISO-19100-Serie umfasst auch Datenmodelle für Zeitabschnitte (ISO 19108), Metadaten (ISO 19115), Dienste (ISO 19119), und anderes mehr. Allerdings sind viele dieser Normen noch im Entwurfsstadium, und es besteht noch wenig Erfahrung, wie tauglich sie in der Praxis sind.
- ◆ **National** – In der Schweiz existieren für mehrere Anwendungsbereiche standardisierte Modelle. Teilweise ist ihre Verwendung in Gesetzen und Verordnungen vorgeschrieben. Für die amtliche Vermessung, für Leitungsnetze und seit kurzem auch für geo-codierte Adressen wurden landesweit gültige Modelle ausgearbeitet.
- ◆ **Branchenverbände** – Eine weitere Quelle für Datenmodelle, auf denen eigene Anwendungen aufbauen können, sind die Branchenverbände. So hat der Schweizerische Ingenieur- und Architektenverband SIA ein INTERLIS-Modell für den Leitungskataster (Abwasser, Gas, Wasser, Fernwärme, Elektrizität, Telecom, Kabelkommunikation) ausgearbeitet.
- ◆ **INTERLIS** – Für einige wenige, sehr grundlegende Bereiche gibt INTERLIS Standardmodelle vor. So wird mit den INTERLIS-Werkzeugen unter anderem ein Modell mit Einheiten (units.ili) ausgeliefert, in dem Meter, Newton, Dezibel und ähnliches definiert werden.

Die Schweizer Modelle werden nach Möglichkeit mit den ISO-Normen in Einklang gebracht. Beispielsweise wurde ein nationales Modell für Metadaten zu geographischen Daten (z.B. Herkunft, Qualität etc.) entsprechend ISO 19115 ausgearbeitet. Das Modell liegt als INTERLIS-Beschreibung vor.

Dank der Verfeinerungs-Mechanismen von INTERLIS kann eine konkrete Anwendung auf ein bereits bestehendes Datenmodell zurückgreifen, ohne dass es *tel quel* übernommen werden müsste.

3.4 Formate zum Datenaustausch

Alle Daten müssen irgendwann in einem bestimmten Format gespeichert werden, damit sie gelagert und zwischen Softwarepaketen ausgetauscht werden können. Neuere Austauschformate sind alle auf der *Extensible Markup Language* (XML) aufgebaut, einem Format, das wegen seinem extrem einfachen Aufbau, der Verwandtschaft mit HTML (dem

Format von Webseiten) und der Abstammung von SGML (einem früher häufig benutzten Format) grosse Verbreitung gefunden hat.

Beispielsweise ist die *Geographic Markup Language* (GML) ein fest definierter XML-Dialekt, der von OpenGIS, einem Herstellerkonsortium von GIS-Software, propagiert wird. Dagegen schreibt INTERLIS keinen bestimmten XML-Dialekt vor, sondern gibt Regeln, anhand derer das Austauschformat automatisch aus dem jeweiligen Datenmodell abgeleitet wird.

An sich ist die Frage des Datenformats sekundär: Solange zu übertragende Daten auf ähnlichen Modellen aufbauen, ist es mit geringem Aufwand möglich, die Daten von einem Format in ein anderes umzuwandeln.

3.5 Wichtige Informationsquellen

Weitere Informationen können unter anderem von folgenden Quellen bezogen werden:

- ◆ **www.interlis.ch** – Das Internet-Portal zu INTERLIS verweist auf Unterlagen, Kurse usw. im Zusammenhang mit INTERLIS. Von dieser Adresse können auch die Spezifikation des Standards und der vorliegende Text elektronisch bezogen werden. Schliesslich sind auf dieser Webseite auch Werkzeuge verfügbar. Zur Zeit sind dies ein Computerprogramm, mit dem INTERLIS-Modelle in der graphischen UML-Notation editiert werden können («UML-Editor»), und eine Softwarebibliothek zum Einlesen, Prüfen, Verändern und Ausgeben von INTERLIS-Modellen («INTERLIS-Compiler»).
- ◆ **www.omg.org** – Von der Webseite der Object Management Group kann die Definition der Unified Modeling Language bezogen werden. Ausserdem sind dort allgemeine Unterlagen (wie Schulungsmaterial oder Presstexte) zum Modellbasierten Ansatz zu finden.
- ◆ **www.w3.org** – Das *World Wide Web Consortium* verwaltet unter anderem auch die Spezifikation von XML.

4. Das Beispiel Ilistal konkret

4.1 Das Datenmodell im Überblick

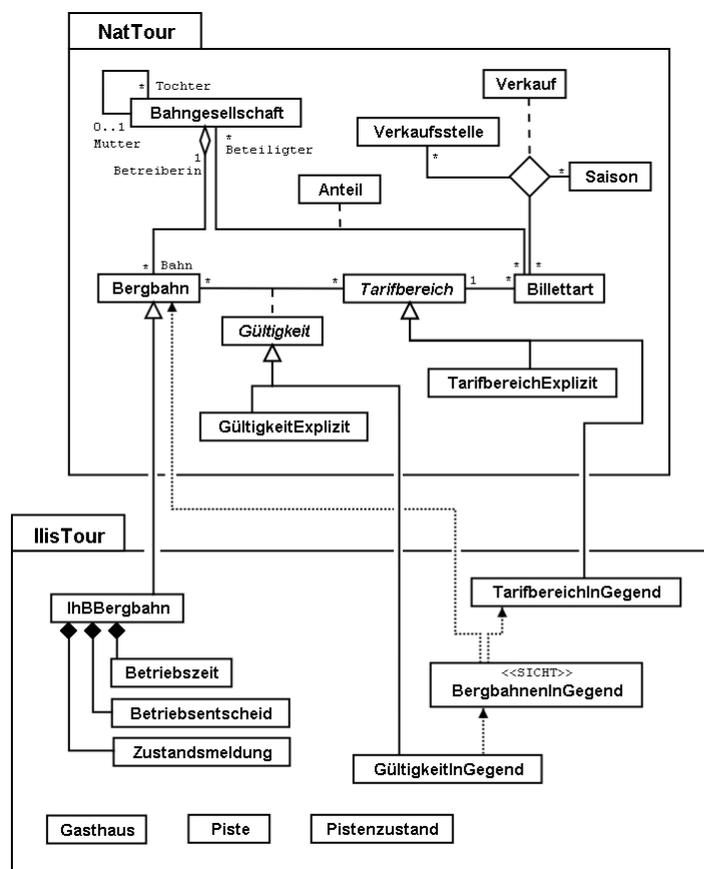


Abbildung 21: Das Datenmodell in der UML-Darstellung. Der Übersichtlichkeit halber sind Themen, Strukturen, Wertebereiche und Attribute nicht abgebildet. Hierzu wird auf Kapitel 6 verwiesen, das auf Einzelheiten näher eingeht.

4.2 Das Datenmodell in INTERLIS

Nachfolgend sind das nationale Grundlagenmodell, das Modell des nationalen Tourismusverbandes sowie das Ilistaler Tourismus-Modell vollständig abgedruckt. Die Modelle sind ausgiebig kommentiert. Zudem verweisen Kästchen auf die Abschnitte im Text, in welchem ausführlicher auf ein Thema eingegangen wird.

Es wird empfohlen, das Beispiel-Datenmodell nur grob zu überfliegen und dann Kapitel 5 bis 8 zu lesen, wo verschiedene Aspekte dieses Datenmodells im Detail besprochen werden.

4.2.1 Ahland.ili – Das nationale Grundlagenmodell

```
!! Das nachfolgende Modell ist in der Version 2.3 von INTERLIS geschrieben.
INTERLIS 2.3;
```

```
!! Das nationale Grundlagen-Modell umfasst keine Daten, sondern definiert
!! lediglich einige Typen und Einheiten, damit sich andere Modelle darauf
!! beziehen koennen.
```

```
TYPE MODEL Ahland AT "http://www.interlis.ch/models/ahland"
  VERSION "2008-01" =
```

```
IMPORTS UNQUALIFIED INTERLIS;
!! Das nationale Grundlagen-Modell baut auf folgenden
!! Grundlagen-Modellen von INTERLIS auf.
!! Units: Einheiten wie beispielsweise Grad Celsius
!! CoordSys: Grundlagen für Koordinatensysteme
IMPORTS Units;
IMPORTS CoordSys;
```

```
!! Das Ilistal liegt in Ahland, in welchem mit Talern bezahlt wird.
!! Der Taler ist eine Einheit (UNIT) fuer Geld (MONEY). Zwar definiert
!! das Einheiten-Modell (Units) bereits einige Waehrungen - beispielsweise
!! Schweizerfranken, Euro und US-Dollar -, nicht aber den Ahlaender Taler.
```

```
UNIT
  Taler EXTENDS MONEY;
```

Einheiten → 6.1.2

```
!! Die Zeit wird in Ahland gemäss der lokalen Zeitzone Ahlandzeit
!! festgehalten.
```

```
REFSYSTEM BASKET TimeSystems ~ INTERLIS.TIMESYSTEMS
  OBJECTS OF TIMEOFDAYSYS: AZ; !! Ahlandzeit
```

```
!! Das Landeskoordinatensystem von Ahland.
REFSYSTEM BASKET CoordSystems ~ CoordSys.CoordsysTopic
  OBJECTS OF GeoCartesian2D: AhlandSys
  OBJECTS OF GeoHeight: AhlandHoechenSys;
```

Koord.-Systeme → 6.7.3

```
DOMAIN
```

```
LandesKoord = COORD 500.00 .. 91000.00 [m] {CoordSystems.AhlandSys[1]},
              700.00 .. 23000.00 [m] {CoordSystems.AhlandSys[2]},
              ROTATION 2 -> 1;
```

Wertebereiche → 6.6

```
LandesKoord3 = COORD 500.00 .. 91000.00 [m] {CoordSystems.AhlandSys[1]},
               700.00 .. 23000.00 [m] {CoordSystems.AhlandSys[2]},
               -200.00 .. 14000.00 [m] {CoordSystems.AhlandHoechenSys[1]},
               ROTATION 2 -> 1;
```

Koordinaten → 6.7

3D-Koordinaten → 6.7.5

```
STRUCTURE Zeitdauer (ABSTRACT) =
END Zeitdauer;
```

Zeitdauer → 6.12

```
STRUCTURE ZeitdauerImplizit EXTENDS Zeitdauer =
  Dauer: MANDATORY (Tag, Woche, Monat, Jahr);
END ZeitdauerImplizit;
```

Vererbung → 5

```
STRUCTURE ZeitdauerExplizit (ABSTRACT) EXTENDS Zeitdauer =
  Dauer (ABSTRACT): MANDATORY NUMERIC [TIME];
END ZeitdauerExplizit;
```

```
STRUCTURE ZeitdauerInMinuten EXTENDS ZeitdauerExplizit =
  Dauer (EXTENDED): MANDATORY 0 .. 200 [Units.min];
END ZeitdauerInMinuten;
```

```
STRUCTURE ZeitdauerInTagen EXTENDS ZeitdauerExplizit =
  Dauer (EXTENDED): MANDATORY 0 .. 1000 [Units.d];
END ZeitdauerInTagen;
```

```
STRUCTURE Zeitpunkt (ABSTRACT) =
END Zeitpunkt;
```

Zeitpunkte → 6.12.5

```
STRUCTURE AhlandTageszeit EXTENDS INTERLIS.TimeOfDay =
  Hours (EXTENDED): 0..23 CIRCULAR [INTERLIS.h];
END AhlandTageszeit;
```

DOMAIN

```
AhlandXMLTageszeit = FORMAT BASED ON AhlandTageszeit
  ( Hours/2 ":" Minutes ":" Seconds );
```

END Ahland.

4.2.2 Adressen.ili – Das Modell für Gebäudeadressen

Das Modell für Gebäudeadressen wird nicht abgedruckt, denn es würde mehrere Seiten umfassen und ist im Zusammenhang des Ilistals nicht im Detail von Interesse. Das Ilistaler Tourismus-Modell etabliert jedoch eine Beziehung zwischen Gasthäusern und den im Adressmodell definierten Hauseingängen.

```
INTERLIS 2.3;
```

Bezug auf Gebäude-
adressen → 2.3.4

```
MODEL Adressen AT "http://www.interlis.ch/models/ahland"
  VERSION "2008-01" =
```

```
TOPIC Gebaeude =
```

```
  CLASS Hauseingang =
    !! ...
  END Hauseingang;
```

```
END Gebaeude;
```

END Adressen.

4.2.3 NatTour.ili – Das Modell des Nationalen Tourismus-Verbands

INTERLIS 2.3;

CONTRACTED MODEL NatTour AT "http://www.interlis.ch/models/ahland"
VERSION "2008-01" =

!! Das Modell des nationalen Tourismus-Verbands baut seinerseits
!! auf dem nationalen Grundlagen-Modell von Ahland auf.
IMPORTS Units, CoordSys, Ahland;

Vorhandene Modelle
nutzen → 6.16

FUNCTION Multiply(factor1 : NUMERIC; factor2 : NUMERIC) : NUMERIC;

!! Eine Bezeichnung umfasst einen Namen und
!! zusaetzlich die Sprache, in welcher der Name
!! geschrieben ist.

Mehrsprachige Objekt-
eigenschaften → 6.11

STRUCTURE Bezeichnung =

!! Der Name darf beliebig lang sein.

Name: TEXT;

!! Der aus zwei Zeichen bestehende Sprachcode nach ISO 639.

!! Beispiele: de = Deutsch, fr = Franzoesisch,

!! it = Italienisch, rm = Raetoromanisch, en = Englisch.

Sprache: TEXT*2;

END Bezeichnung;

Zeichenketten → 6.4

TOPIC Bergbahnen =

!! Eine Bahnbezeichnung ist wie eine gewoehnliche
!! Bezeichnung (aber mit hoechstens 100 Zeichen),
!! umfasst aber zusaetzlich noch eine Kurzform des Namens,
!! zum Beispiel "IhB" fuer die Ilishornbahnen.

Sprachabhaengige Be-
zeichnungen als Struk-
turelemente → 6.11.2

STRUCTURE Bahnbezeichnung EXTENDS Bezeichnung =

Name (EXTENDED): TEXT*100;

Kurzbezeichnung: TEXT*10;

END Bahnbezeichnung;

!! Eine Bahngesellschaft betreibt Bahnen.

CLASS Bahngesellschaft =

!! Die Namen dieser Bahngesellschaft, allenfalls in unterschiedlichen

!! Sprachen. Es muss mindestens ein (1) Name bekannt sein, jedoch

!! gibt es nach oben keine (*) Beschraenkung der Anzahl der Namen.

Namen: BAG {1..*} OF Bahnbezeichnung;

!! Es soll pro Sprache nur eine einzige Bahnbezeichnung

!! geben koennen: Die Ilishornbahnen duerfen somit nur

!! eine einzige italienische Bezeichnung besitzen.

!! Allerdings gilt diese Einschraenkung nur lokal, also

!! pro Bahngesellschaft. Auch den Blaubergbahnen soll es

!! ja gestattet sein, einen italienischen Namen zu tragen.

UNIQUE

(LOCAL) Namen : Sprache;

END Bahngesellschaft;

Eindeutigkeitsbedingungen
→ 6.14.3

CLASS Bergbahn =

!! Die Namen dieser Bergbahn, allenfalls in unterschiedlichen

!! Sprachen. Es muss mindestens ein (1) Name bekannt sein, jedoch

```

!! gibt es nach oben keine Beschraenkung (*) der Anzahl der Namen.
Namen: BAG {1..*} OF Bezeichnung;
LageTalstation: Ahland.LandesKoord;
LageBergstation: Ahland.LandesKoord;
Fahrzeit: Ahland.ZeitdauerInMinuten;
!! Die genaue Art der Bergbahn.
Art: (Zahnradbahn,
      Standseilbahn,
      Lufseilbahn,
      Skilift,
      Sessellift,
      Gondelbahn);
END Bergbahn;

ASSOCIATION =
!! Gibt an, welche Bahnen eine konkrete Bahngesellschaft
!! betreibt. Beispiel: Die "Ilishornbahnen" betreiben die
!! Standseilbahn "Ilisdorf-Ilishorn", die Gondelbahn
!! "Ilisbad-Ilisegg" und den Skilift "Ilisegg-Ilishorn".
!! Eine Bahngesellschaft kann beliebig viele {*} Bergbahnen
!! betreiben und es gibt immer genau eine {1} Betreiberin
!! je Bahn.
!! Das Zeichen -- steht fuer eine gewoehnliche
!! Beziehung, -<> besagt, dass die Beziehung etwas
!! staerker als gewoehnlich ist, naemlich eine
!! sogenannte Aggregation.
Betreiberin -<> {1} Bahngesellschaft;
Bahn -- {*} Bergbahn;
END;

ASSOCIATION =
Tochter -- {*} Bahngesellschaft;
Mutter -- {0..1} Bahngesellschaft;
END;

END Bergbahnen;

TOPIC Billette =
DEPENDS ON Bergbahnen;
!! Die national definierten impliziten Zeitdauern sind
!! Tag, Woche, Monat und Jahr. Bei Billetten gibt es
!! eine weitere implizite Zeitdauer, naemlich die Saison
!! (fuer Saisonpaesse).

STRUCTURE ZeitdauerImplizit EXTENDS Ahland.ZeitdauerImplizit =
  Dauer (EXTENDED): (Saison);
END ZeitdauerImplizit;

!! Ein Bereich, in dem eine bestimmte Billettart gueltig
!! ist.
CLASS Tarifbereich (ABSTRACT) =
END Tarifbereich;

CLASS TarifbereichExplizit EXTENDS Tarifbereich =
END TarifbereichExplizit;

!! Eine Art Billet, zum Beispiel der "Ilosaurus-Wochenpass".

```

Koordinaten → 6.7

Arten → 6.2

Stärke der Beziehung
→ 6.13.2

Rollennamen → 6.13.1

Zeitdauer → 6.12

Abstrakte Klassen → 5.3

```

CLASS Billettart =
  !! Die Namen dieser Billettart, allenfalls in unterschiedlichen
  !! Sprachen. Es muss mindestens ein (1) Name bekannt sein, jedoch
  !! gibt es nach oben keine Beschaenkung (*) der Anzahl der Namen.
  Namen: BAG {1..*} OF Bezeichnung;
  !! Der Preis eines Billetts in Talern. Die Waehrung
  !! wird im nationalen Basismodell von Ahland definiert.
  Preis: MANDATORY 0.00 .. 9999.99 [Ahland.Taler];
  !! Die Gueltigkeitsdauer eines Billets. Sie kann explizit sein,
  !! z.B. fuer Billette, die 120 Minuten gueltig sind, oder
  !! implizit, beispielsweise fuer Wochen- oder Saisonpaeese.
  Gueltigkeitsdauer: MANDATORY Ahland.Zeitdauer;
END Billettart;

ASSOCIATION =
  Tarifbereich -- {1} Tarifbereich;
  Billettart -- {*} Billettart;
END;

ASSOCIATION Gueltigkeit (ABSTRACT) =
  Bergbahn (EXTERNAL) -- {*} NatTour.Bergbahnen.Bergbahn;
  Tarifbereich -- {*} Tarifbereich;
END Gueltigkeit;

!! Eine Beziehung zwischen Bergbahn und Tarifbereich,
!! die nicht abgeleitet, sondern manuell eingegeben wurde.
ASSOCIATION GueltigkeitExplizit EXTENDS Gueltigkeit =
  Tarifbereich (EXTENDED) -- TarifbereichExplizit;
END GueltigkeitExplizit;

ASSOCIATION Anteil =
  Beteiligter (EXTERNAL) -- {*} NatTour.Bergbahnen.Bahngesellschaft;
  Billettart -- {*} Billettart;
ATTRIBUTE
  Anteil: 0.0 .. 100.0 [Units.Percent];
END Anteil;

CLASS Verkaufsstelle =
  Namen: BAG {1..*} OF Bezeichnung;
END Verkaufsstelle;

CLASS Saison =
  Anfang: FORMAT INTERLIS.XMLDate "1900-1-1" .. "2299-12-31";
  Ende: FORMAT INTERLIS.XMLDate "1900-1-1" .. "2299-12-31";
END Saison;

ASSOCIATION Verkauf =
  Verkaufsstelle -- {*} Verkaufsstelle;
  Saison -- {*} Saison;
  Billettart -- {*} Billettart;
ATTRIBUTE
  Anzahl: 1 .. 999999 [Units.CountedObjects];
  Betrag: 0.00 .. 9999999.99 [Ahland.Taler]
  := Multiply(Anzahl, Billettart -> Preis);
END Verkauf;

END Billette;

```

Numerische Datentypen
→ 6.1

Beziehungsklassen
→ 6.13.3

Mehrgliedrige
Beziehungsklassen
→ 6.13.4

```
END NatTour.
```

4.2.4 IlistTour.ili – Das Ilistaler Tourismus-Modell

```
INTERLIS 2.3;
```

```
CONTRACTED MODEL IlistTour AT "http://www.interlis.ch/models/ahland"
  VERSION "2008-01" =
```

```
!! Um dieses Modell zu implementieren, muss ein Programmpaket
!! die Funktion AhlandToWGS84 unterstuetzen. Dies kann nicht
!! einfach vorausgesetzt werden, sondern ist vertraglich mit
!! dem Hersteller zu vereinbaren. Die Notwendigkeit eines solchen
!! Kontrakts wird mit CONTRACTED angemerkt.
```

Kontrakte → 7.2

```
IMPORTS UNQUALIFIED INTERLIS;
IMPORTS Units, CoordSys, Ahland, Adressen, NatTour;
```

```
!! Touristen mit einfachen GPS-Empfaengern soll ein besonderer Service
!! geboten werden. Ihre Empfaenger zeigen Koordinaten im Koordinatensystem
!! WGS84 an. Es arbeitet mit Winkeln in Grad, Minuten und Sekunden; die
!! entsprechende Winkel-Einheit ist im INTERLIS-Einheitenmodell bereits
!! definiert.
```

```
REFSYSTEM BASKET CoordSystems ~ CoordSys.CoordsysTopic
  OBJECTS OF GeoEllipsoidal: WGS84
  OBJECTS OF GeoHeight: WGS84H;
```

```
DOMAIN
```

```
WGS84Koord = COORD -90.00000 .. 90.00000 [Units.Angle_Degree] {WGS84[1]},
              0.00000 .. 359.99999 CIRCULAR [Units.Angle_Degree]
              {WGS84[2]},
              -2000.00 .. 9000.00 [m] {WGS84H[1]};
```

Geometrietypen → 6.9

```
AhlandLinie (ABSTRACT) = POLYLINE VERTEX Ahland.LandesKoord;
AhlandLinieNormal EXTENDS AhlandLinie = POLYLINE WITH (STRAIGHTS, ARCS);
AhlandLinieGerichtet EXTENDS AhlandLinieNormal = DIRECTED POLYLINE;
AhlandFlaechen = SURFACE WITH (STRAIGHTS, ARCS) VERTEX Ahland.LandesKoord
  WITHOUT OVERLAPS > 0.02;
AhlandGebietseinteilung EXTENDS AhlandFlaechen = AREA;
```

```
!! Die Umrechnung von Ahlaendern Landeskoordinaten zu WGS84.
FUNCTION AhlandToWGS84 (Ah: Ahland.LandesKoord): WGS84Koord;
FUNCTION InSurface (Lage: Ahland.LandesKoord;
  Gegend: AhlandFlaechen): BOOLEAN;
```

Funktionen → 7.2

```
TOPIC IhBBergbahnen EXTENDS NatTour.Bergbahnen =
```

```
CLASS IhBBergbahn EXTENDS NatTour.Bergbahnen.Bergbahn =
  !! Im Ilistal gibt es neben den national
  !! ueblichen Bergbahn-Arten auch den Schneeбус.
  Art (EXTENDED): (Schneeбус);
  !! Der nationale Tourismusverband interessiert sich nicht fuer
  !! die Hoehen. In einem Wintersportgebiet wie dem Ilistal sind
  !! sie aber von grosser Bedeutung. Daher werden die Lagen im
  !! Ilistal als dreidimensionale Koordinaten (inklusive Hoehen)
```

Aufzählungen erweitern
→ 6.3.2

```

!! erfasst, sind also im Vergleich zum nationalen Modell erweitert.
LageTalstation (EXTENDED): Ahland.LandesKoord3;
LageBergstation (EXTENDED): Ahland.LandesKoord3;
LageTalstationWGS: WGS84Koord := AhlandToWGS84(LageTalstation);
LageBergstationWGS: WGS84Koord := AhlandToWGS84(LageBergstation);
!! Manche Bahnen haben eine Web-Kamera installiert, die laufend die
!! Umgebung der Bergstation aufnimmt, damit interessierte Touristen
!! sehen, ob sich die Reise lohnt. Der Eintrag zur Bergbahn besagt
!! ueber einen Uniform Resource Identifier (URI, eine
!! Adresse auf dem Internet), wo das aktuelle Bild
!! verfuegbar ist.
BildBergstation: URI;
Trasseeverlauf: AhlandLinieNormal;
WandererSchlittler: (ungeeignet, geeignet);
END IhBBergbahn;

VIEW CheckTrasseStartAndEndPoint
  INSPECTION OF Trasse ~ IhBBergbahn -> Trasseeverlauf;
=
MANDATORY CONSTRAINT
  !! Der erste Punkt des Trasseeverlaufs muss die Tal-,
  !! der letzte Punkt die Bergstation sein.
  Trasse -> Segments[FIRST] -> SegmentEndPoint == PARENT -> LageTalstation
  AND
  Trasse -> Segments[LAST] -> SegmentEndPoint == PARENT -> LageBergstation;
END CheckTrasseStartAndEndPoint;

!! Ein besonderer Tarifbereich, an dem alle Bahnen teilnehmen,
!! die in einer raeumlich umgrenzten Gegend liegen.
CLASS TarifbereichInGegend EXTENDS NatTour.Billette.Tarifbereich =
  Gegend: AhlandFlaeche;
END TarifbereichInGegend;

!! Eine Sicht, die alle Bahnen umfasst, deren Tal- und Bergstation
!! innerhalb der Gegend eines Tarifsbereichs liegt. Natuerlich koennen
!! nur jene Tarifbereiche beruecksichtigt werden, die als Gegend beschrieben
!! sind (TarifbereichInGegend); ein expliziter Tarifbereich wuerde hier
!! keinen Sinn machen.
VIEW BergbahnenInGegend
JOIN OF Bb ~ NatTour.Bergbahnen.Bergbahn,
      T ~ TarifbereichInGegend;
WHERE InSurface(Bb -> LageTalstation, T -> Gegend) AND
      InSurface(Bb -> LageBergstation, T -> Gegend);
=
END BergbahnenInGegend;

!! Eine Beziehung zwischen Billettart und Tarifbereich,
!! die nicht manuell eingegeben, sondern automatisch
!! aufgrund der Lage von Tal- und Bergstation abgeleitet
!! wurde.
ASSOCIATION GueltigkeitInGegend EXTENDS NatTour.Billette.Gueltigkeit
DERIVED FROM BiG ~ BergbahnenInGegend
=
  Bergbahn (EXTENDED) -- Bergbahn := BiG -> Bb;
  Tarifbereich (EXTENDED) -- TarifbereichInGegend := BiG -> T;
END GueltigkeitInGegend;

END IhBBergbahnen;

```

Internet-Adressen → 6.4

Aus organisatorischen
Gründen erben → 5.6Konsistenzbedingungen
→ 6.14

Sichten → 6.17

Ableitbare Beziehungen
→ 6.13.7

```
TOPIC Gasthaeuser =
  DEPENDS ON Adressen.Gebaeude;

  CLASS Gasthaus =
    !! Die Namen dieses Gasthauses, allenfalls in unterschiedlichen
    !! Sprachen. Es muss mindestens ein (1) Name bekannt sein, jedoch
    !! gibt es nach oben keine Beschraenkung (*) der Anzahl der Namen.
    Namen: BAG {1..*} OF NatTour.Bezeichnung;
    !! Die Internet-Adresse (Uniform Resource Identifier,
    !! abgekuerzt URI) eines Fotos des Gasthauses.
    Bild: URI;
  END Gasthaus;

  !! Die Ilistaler definieren nicht selbst, was sie unter einer Adresse
  !! verstehen. Stattdessen etablieren sie eine Beziehung zwischen Gasthaus
  !! und seinem Eingang. Dadurch koennen sie die Koordinaten der Gasthaeuser
  !! aus den Daten der amtlichen Vermessung uebernehmen und brauchen sie
  !! nicht selber zu erfassen.
  ASSOCIATION =
    Gasthaus -- Gasthaus;
    Eingang (EXTERNAL) -- Adressen.Gebaeude.Hauseingang;
  END;

END Gasthaeuser;

TOPIC IhBPlanung =
  DEPENDS ON IlistTour.IhBBergbahnen;

  CLASS Betriebszeit =
    Startdatum: INTERLIS.XMLDate;
    Beginn: Ahland.AhlandXMLTageszeit;
    Schluss: Ahland.AhlandXMLTageszeit;
  END Betriebszeit;

  ASSOCIATION =
    Bahn (EXTERNAL) -<#> {1} IlistTour.IhBBergbahnen.IhBBergbahn;
    Betriebszeit -- {*} Betriebszeit;
  END;

END IhBPlanung;

TOPIC IhBBetrieb =
  DEPENDS ON IlistTour.IhBBergbahnen;

  CLASS Betriebsentscheid =
    Zeitpunkt: INTERLIS.XMLDateTime;
    Entscheid: (ja, nein);
  END Betriebsentscheid;

  ASSOCIATION =
    Bahn (EXTERNAL) -<#> {1} IlistTour.IhBBergbahnen.IhBBergbahn;
    Betriebsentscheid -- {*} Betriebsentscheid;
  END;
```

[Internet-Adressen → 6.4](#)[Komposition → 6.13.2](#)

```

END IhBBetrieb;

TOPIC IhBAktuell =
  DEPENDS ON IlisTour.IhBBergbahnen;

  STRUCTURE Windangabe =
    Windrichtung: MANDATORY (N, NE, E, SE, S, SW, W, NW) CIRCULAR;
    Windgeschwindigkeit: MANDATORY 0 .. 200 [Units.kmh];
  END Windangabe;

  CLASS Zustandsmeldung =
    !! Die Temperatur ist in Grad Celsius angegeben. Diese
    !! Einheit wird vom INTERLIS-Einheitenmodell (Units)
    !! definiert. MANDATORY besagt, dass die Temperatur
    !! bekannt sein muss.
    Temperatur: MANDATORY -50 .. 50 [Units.oC];
    !! Das Attribut Wind bezieht sich auf obige Struktur
    !! Windangabe.
    Wind: Windangabe;
    Wartezeit: Ahland.ZeitdauerInMinuten;
    Erfasst: MANDATORY INTERLIS.XMLDateTime;
  END Zustandsmeldung;

  ASSOCIATION =
    Bahn (EXTERNAL) -<#> {1} IlisTour.IhBBergbahnen.IhBBergbahn;
    Zustandsmeldung -- {*} Zustandsmeldung;
  END;

END IhBAktuell;

TOPIC Pisten =

  CLASS Piste =
    Schwierigkeitsgrad: (blau, rot, schwarz: FINAL) ORDERED;
    Verlauf: AhlandLinieGerichtet;
  END Piste;

END Pisten;

TOPIC Pistenzustaende =

  CLASS Pistenzustand =
    PraeparierteFlaeche: AhlandGebietseinteilung;
  END Pistenzustand;

END Pistenzustaende;

END IlisTour.

```

Strukturen
→ 6.10

Zyklische Aufzählungen
→ 6.3.1

Fakultative und obligato-
rische Attribute → 6.5

Aufzählungen → 6.3.1

Linien → 6.9

Gebietseinteilung → 6.9.4

4.3 Transferdaten

Wollen die Ilistaler ihre gesamten Daten an den nationalen Tourismusverband schicken, erstellen sie (mit ihrem Softwarepaket) eine Transferdatei. Diese wird zwar normalerweise

von einem anderen Computersystem eingelesen und muss nicht in dieser Form von einem Menschen betrachtet werden. Dennoch ist ein kleiner Teil der Transferdatei nachfolgend abgedruckt, um eine Vorstellung von ihrem Aufbau zu vermitteln.

Drei Punkte (...) zeigen die Auslassungen an; die Kästchen rechts sind lediglich Anmerkungen und gehören nicht zur Transferdatei.

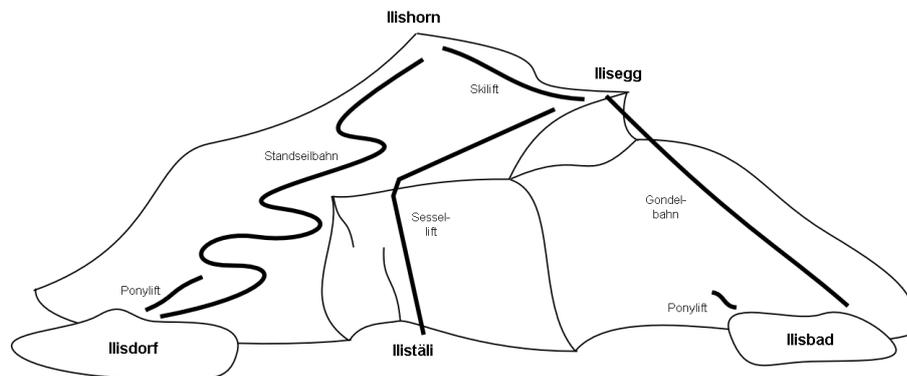


Abbildung 22: Die Bergbahnen auf das Ilishorn sind ein Teil der Daten, die in einer Transferdatei enthalten sind (Wiederholung von Abbildung 11). Die nachfolgende Datei enthält einige Daten für den Ponylift Ilistdorf.

```
<?xml version="1.0" encoding="utf-8"?>
<TRANSFER xmlns="http://www.interlis.ch/INTERLIS2.3">

<HEADERSECTION VERSION="2.3" SENDER="AHTOUIHB0">
  <ALIAS>...</ALIAS>
</HEADERSECTION>

<DATASECTION>
<BASKET BID="xAHTOUIHB01234567" TOPICS="IliTour.IhBBergbahnen">
  <IliTour.IhBBergbahnen.IhBBergbahn TID="xAHTOUIHB04231336">
    <Namen>
      <NatTour.Bezeichnung>
        <Name>Ponylift Ilistdorf</Name>
        <Sprache>de</Sprache>
      </NatTour.Bezeichnung>
    </Namen>
    <LageTalstation>
      <P>
        <C1>7931.11</C1>
        <C2>13171.23</C2>
        <C3>1771.34</C3>
      </P>
    </LageTalstation>
    <LageBergstation>
      <P>
        <C1>8020.60</C1>
        <C2>13188.62</C2>
        <C3>1789.04</C3>
      </P>
    </LageBergstation>
  </IliTour.IhBBergbahn>
</BASKET>
</DATASECTION>
```

Vorspann

Namen

Lage der Talstation

Lage der Bergstation

Fahrzeit

```

<Fahrzeit>
  <Ahland.ZeitdauerInMinuten>
    <Dauer>3</Dauer>
  </Ahland.ZeitdauerInMinuten>
</Fahrzeit>
<Art>Skilift</Art>
<LageTalstationWGS>
  <P>
    <C1>23.68611</C1>
    <C2>44.20278</C2>
    <C3>1771.34</C3>
  </P>
</LageTalstationWGS>
<LageBergstationWGS>
  <P>...</P>
</LageBergstationWGS>
<BildBergstation>
  http://www.ilishornbahnen.com/webcam?bahn=pony4
</BildBergstation>
<Trasseeverlauf>...</Trasseeverlauf>
<WandererSchlittler>ungeeignet</WandererSchlittler>
<Betriebszeit>...</Betriebszeit>
<Betriebsentscheid>...</Betriebsentscheid>
<Zustandsmeldung>
  <Ilistour.IhBAktuell.Zustandsmeldung>
    <Temperatur>13</Temperatur>
    <Wind>
      <Ilistour.IhBAktuell.Windangabe>
        <Windrichtung>NE</Windrichtung>
        <Windgeschwindigkeit>13</Windgeschwindigkeit>
      </Ilistour.IhBAktuell.Windangabe>
    </Wind>
    <Wartezeit>
      <Ahland.ZeitdauerInMinuten>
        <Dauer>8</Dauer>
      </Ahland.ZeitdauerInMinuten>
    </Wartezeit>
    <Erfasst>2002-11-25T15:11:00</Erfasst>
  </Ilistour.IhBAktuell.Zustandsmeldung>
</Zustandsmeldung>
</IlistTour.IhBBergbahnen.IhBBergbahn>
</BASKET>
</DATASECTION>
</TRANSFER>

```

Art

Lage Talstation (WGS84)

Bild von der Bergstation

Zustandsmeldung

5. Erben wird Mode

5.1 Erbstücke und Erbrecht – Prinzipien der Vererbung

Eigentlich ist eine Bergbahn nichts Besonderes, denn sie besitzt viele Eigenschaften, die allen Bahnen gemeinsam sind. Zum Beispiel hat sie wie alle Bahnen einen Namen. Auch dass eine Bergbahn von einer Gesellschaft betrieben wird, ist nicht so speziell – auch bei einer Tramlinie ist das schliesslich nicht anders.

Andererseits ist es offensichtlich, dass Bergbahnen und Tramlinien zwar einiges gemeinsam haben, aber doch nicht ganz dasselbe sind.

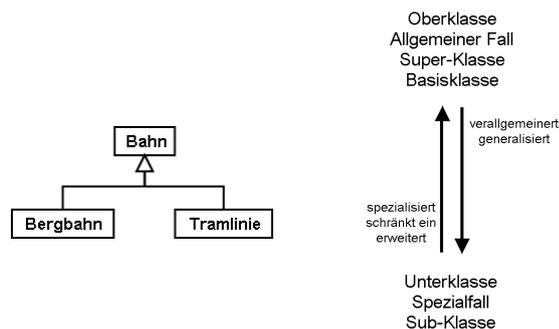


Abbildung 23: In vielem ähnlich, aber doch nicht ganz dasselbe: Bergbahn und Tramlinie sind beides spezielle Bahnen – sie sind Unterklassen der allgemeineren Oberklasse Bahn.

- ▶ Mit der **Vererbung** lassen sich die Gemeinsamkeiten und Unterschiede von Objektklassen formulieren. **Unterklassen spezialisieren** die allgemeineren **Oberklassen**.

Es ist üblich, die allgemeinere Oberklasse im Diagramm oberhalb der spezielleren Unterklasse zu zeichnen. Komplizierte Diagramme können jedoch unübersichtlich werden, wenn man sich strikte an dieses Prinzip hält. Entscheidend ist in jedem Fall die Richtung des Pfeils, nicht die Anordnung auf dem Papier.

Jede Bergbahn ist eine Bahn, aber nicht jede Bahn fährt auf einen Berg: Die Menge aller Bergbahnen ist eine Teilmenge der Menge aller Bahnen. Man sagt auch, die Unterklasse Bergbahn sei eine **Einschränkung** der Oberklasse Bahn.

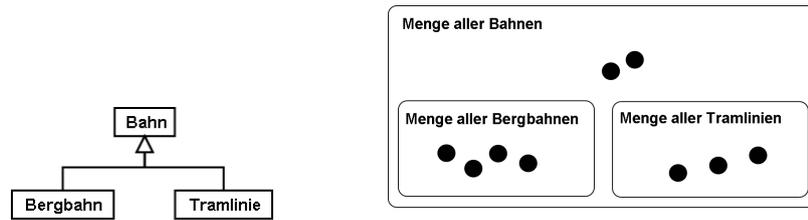


Abbildung 24: Dem Spezialisieren von Klassen entspricht eine Teilmengenbeziehung von Objektmengen: Die Menge aller Bergbahnen (im rechten Bild mit vier Elementen) muss vollständig in der Menge aller Bahnen (neun Elemente) enthalten sein, denn im Modell (linkes Bild) spezialisiert die Klasse Bergbahn die allgemeinere Klasse Bahn.

Gelegentlich wird für die Spezialisierung auch – gleichbedeutend mit «Einschränkung» – der Ausdruck **Erweiterung** gebraucht.

Es ist verwirrend, dass die Begriffe «Einschränkung» und «Erweiterung» beim Modellieren häufig mit gleicher Bedeutung gebraucht werden. Der Grund ist folgender: Eine Klasse kann auch als eine Anzahl von Bedingungen aufgefasst werden, anhand derer sich entscheiden lässt, ob ein Objekt zur Klasse gehört (zum Beispiel Kriterien dafür, wann genau etwas eine Bahn ist). Eine Unterklasse verschärft nun die Anforderungen: Damit etwas eine Bergbahn ist, muss es nicht nur sämtlichen Bedingungen für das Bahn-Sein genügen, sondern es sind zusätzlich weitere Bedingungen zu erfüllen. Indem also eine Unterklasse die Anforderungen erweitert, schränkt sie die Menge der dazugehörigen Exemplare ein.

Die Vererbung ist ein ausgezeichnetes Mittel, um in einer komplexen Welt Ordnung zu schaffen. Allerdings kann es verlockend sein, ein allzu detailliertes Modell zu formulieren – also auch dort Klassen zu unterscheiden, wo es überhaupt nicht nötig wäre.

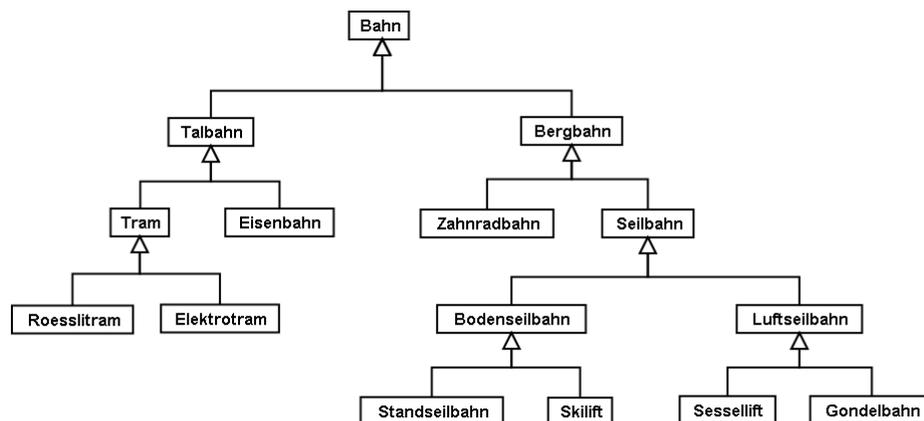


Abbildung 25: Das kraftvolle Mittel der Vererbung kann dazu verleiten, auch dort Spezialfälle zu unterscheiden, wo es von der Anwendung her gar nicht nötig wäre. Zwar ist ein Rösslitrām wirklich nicht ganz dasselbe wie eine elektrische Strassenbahn, aber: Macht es irgendeinen Sinn, all diese Unterscheidungen im Datenmodell zu treffen, oder wird das Ganze so nur unnötig aufgebläht?

Einen Hinweis darauf, ob sich das Unterteilen in spezielle Klassen lohnt, geben die jeweiligen Eigenschaften. So hat jede Bahn einen Namen, aber nur Bergbahnen besitzen eine Tal- und eine Bergstation.

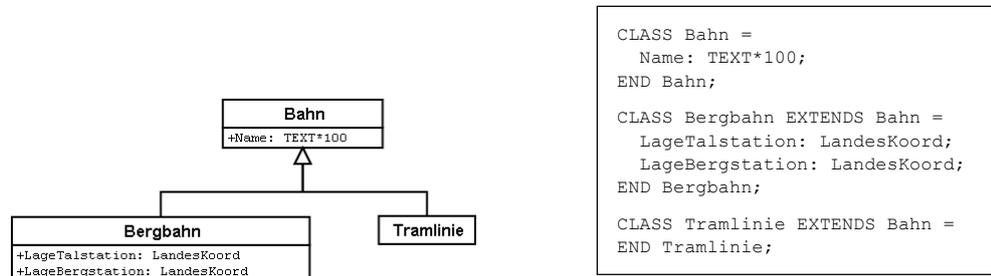


Abbildung 26: Bergbahn und Tramlinie übernehmen («erben») die Eigenschaft Name von ihrer Oberklasse Bahn, ohne dass dies nochmals geschrieben werden müsste. Eine Bergbahn besitzt zusätzlich zu den ererbten noch weitere Eigenschaften, nämlich die Lage von Tal- und Bergstation (in Landeskoordinaten). Rechts dasselbe in der Schreibweise von INTERLIS.

- Unterklassen übernehmen oder **erben** immer alle Eigenschaften von ihren allgemeineren Oberklassen. Sie können jedoch zusätzliche Eigenschaften definieren.

5.2 Ererbtes verfeinern

Hundert Zeichen für den Namen einer Bahn mag im allgemeinen Fall ja angemessen sein. Schliesslich war auch schon einmal die Rede davon gewesen, einen «Kinder- und- Familien-Schlittenlift Region Ilishorn» zu betreiben. Allerdings entschied man sich dann doch noch im letzten Moment für den Namen «Ponylift», zur grossen Erleichterung des Verkehrsvereins.

Aber eine Tramlinie mit hundert Buchstaben? Hier reichen fünf Zeichen definitiv aus.

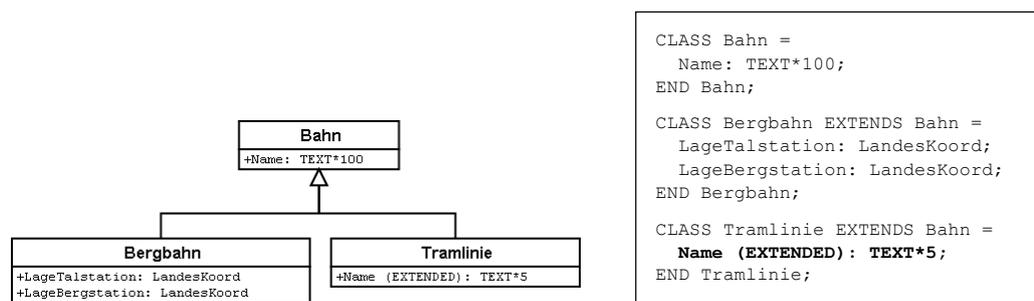


Abbildung 27: Für den Namen einer Tramlinie reichen fünf Zeichen aus: Der Typ der Eigenschaft «Name» wird von der Unterklasse verfeinert (präzisiert). Rechts wiederum dasselbe in der Schreibweise von INTERLIS.

Die präzisierte, verfeinerte Angabe muss mit der übernommenen verträglich sein. Beispielsweise darf die maximal erlaubte Namenslänge bei der Tramlinie nicht länger als bei der allgemeinen Bahn sein.

- ▶ Unterklassen können ererbte Eigenschaften **verfeinern**. Die präzisierten Angaben dürfen aber nicht im Widerspruch zu den übernommenen stehen: sie müssen mit den Definitionen der Oberklasse **verträglich** sein.

Andernfalls könnte es zu einer Unterklasse Objekte geben, die nicht zur Menge aller Objekte der Oberklasse gehören.

5.3 Gibt es das denn überhaupt? – Abstrakte Klassen

Manche Klassen sind rein gedankliche Hilfsmittel: Es gibt von ihnen keine real existierenden Exemplare. Beispielsweise gibt es kein einziges Lebewesen auf der Welt, das nur Lebewesen, nicht aber auch noch etwas Spezielleres wäre. Ebenso könnte ein Datenmodell festlegen, dass es keine Bahn an und für sich geben soll, sondern dass jede Bahn entweder eine Bergbahn, eine Tramlinie etc. sein müsse.

- ▶ Soll es von einer Klasse keine konkreten Objekte geben können, wird sie als **abstrakt** erklärt.

Häufig sind in einem Datenmodell sogar alle Oberklassen abstrakt und nur die untersten, speziellesten Klassen konkret.

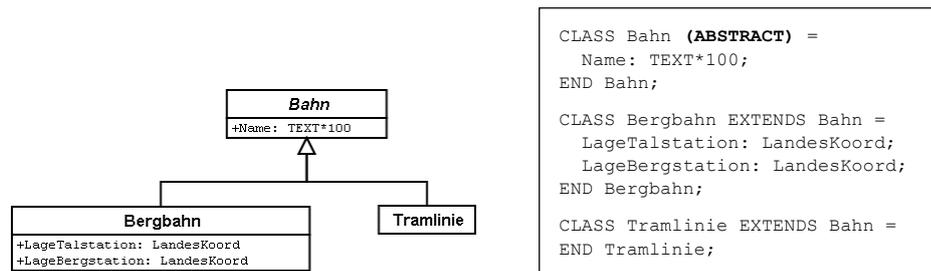


Abbildung 28: Bahn als abstrakte Klasse: Soll es keine Objekte geben können, die nur Bahn sind, ohne auch Bergbahn oder Tramlinie zu sein, wird dies im Diagramm mittels Schrägschrift bezeichnet. Rechts dasselbe Modell in der Schreibweise von INTERLIS.

5.4 So genau wollen wir das nicht vorschreiben – Abstrakte Eigenschaften

Angenommen, ein internationaler Verband möchte sicherstellen, dass Billette mit ihren Preisen erfasst werden. Er will aber keine bestimmte Währung für die Preisangabe vorschreiben, und entsprechend ist auch nicht klar, was eine sinnvolle Obergrenze für den Preis wäre. Fest steht andererseits, dass «Preis» eine Zahl sein soll, und dass es sich um Geld handelt. Schliesslich werden Preise nicht in Kilometern pro Stunde gemessen!

```

CLASS BillettartWeltweit (ABSTRACT) =
  Preis (ABSTRACT): NUMERIC [MONEY];
END BillettartWeltweit;

CLASS BillettartAhland EXTENDS BillettartWeltweit =
  Preis (EXTENDED): 0.00 .. 9999.99 [Ahland.Taler];
END BillettartAhland;

```

- ▶ Nicht alle Eigenschaften müssen bis ins Detail festgelegt werden: Bei abstrakten Klassen sind **abstrakte Eigenschaften** zulässig. Es liegt dann an den konkreten Unterklassen, diese Eigenschaften zu präzisieren. Dies ist zum Beispiel dann nützlich, wenn etwas auf internationaler oder nationaler Stufe allgemein geregelt werden soll, ohne gleich jedes Detail vorzuschreiben.

5.5 Details interessieren nicht – Das Spezielle allgemeiner betrachten

Wer sich allgemein nach den Bahnen im Land erkundigt, interessiert sich nicht dafür, ob es sich bei einem bestimmten Exemplar nun um eine Bergbahn, ein Tram oder sonst irgendeine Unterart von Bahn handelt. Er will auch nicht wissen, welches Stangensystem eine Bahn benutzt, falls es denn eine Zahnradbahn sein sollte. Allein schon der Name (der gemäss Datenmodell für jede Bahn erfasst ist) reicht als Antwort.

- ▶ Exemplare einer Unterklasse können immer auch verallgemeinernd im Sinn einer Oberklasse gesehen werden.

Der griechische Ausdruck für dieses Prinzip heisst Polymorphismus (Vielgestaltigkeit).

Dies funktioniert aber nur unter einer Bedingung:

- ▶ Jede Erweiterung muss mit ihrer Basisdefinition **verträglich** sein. Verträglich heisst, dass jeder Wert, der mit der erweiterten Definition möglich ist, gemäss Regeln des Grundtyps (Text, Aufzählung, Zahl, Koordinate, etc.) auf die Basisdefinition abgebildet werden kann.

5.6 Vererbung im Grossen

Nicht immer ist die Unterscheidung zwischen Ober- und Unterklasse rein sachlich gerechtfertigt. Auch organisatorische Gründe können den Ausschlag geben.

Beispielsweise ist man im Ilistal mit der Vorstellung, die sich der nationale Tourismusverband von einer Bergbahn macht, im Prinzip zwar einverstanden. Ganz zufrieden ist man jedoch nicht:

- ◆ Für die Bahnen aufs Ilishorn wäre es interessant, den Trasseeverlauf zu kennen. Würde er erfasst, könnte man den Verlauf in die Kärtchen einzeichnen, die der Verkehrsverein gratis an Touristen abgibt.
- ◆ Ausserdem möchten die Ilistaler erfassen, welche Bahnen sich für Wanderer und Schlittler eignen.

Beides sind Eigenschaften, die an sich jede Bergbahn tragen kann – nur fehlen sie eben im nationalen Modell. Natürlich haben die Ilistaler den nationalen Verband darum gebeten, sein Modell anzupassen. Aber von dort war nur zu hören, man habe weder die Zeit noch das

Geld, wegen der Extrawürste eines Bergtals im ganzen Land die Computersysteme zu ändern. Was nun?

Die einen fanden, man solle doch den nationalen Verband einfach ignorieren. Die da oben seien ja doch nur ein Haufen von Bürokraten, ohne jegliches Verständnis für die Anliegen vor Ort! (Es fielen auch noch andere Worte, die aber nichts zur Sache tun.)

Andere konnten die Sicht des nationalen Verbands durchaus verstehen – wenn da jedes Täli kommen würde! Und ausserdem würde man ja doch auch vom nationalen Verband profitieren: Mit den Daten, die man ihm schickt, wird schliesslich auch Material für und über das Ilistal produziert.

Sollten die Ilistaler also auf ihre Sonderwünsche verzichten? Oder alle Daten doppelt erfassen – einmal für sich selber, einmal für den nationalen Verband?

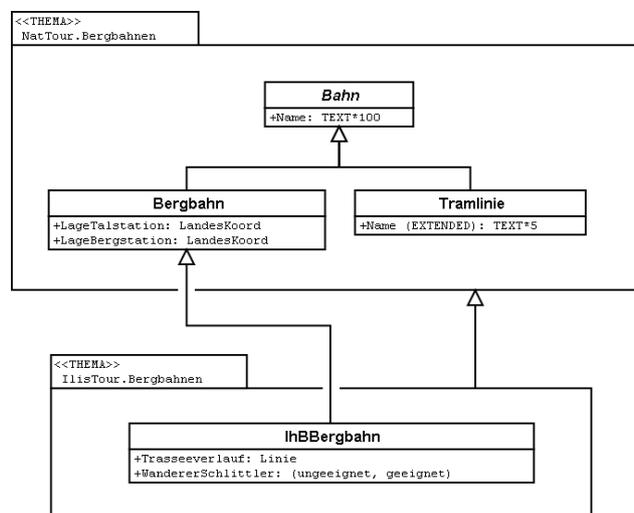


Abbildung 29: Der nationale Tourismusverband ist nicht bereit, sein Modell an die Ilistaler Spezialwünsche anzupassen. Dank der Vererbung können die Ilistaler dennoch ihre Daten erfassen: Ihr Thema Bergbahnen übernimmt alles vom nationalen Thema Bergbahnen, erweitert es aber um eine Objektklasse IhBBergbahn mit zusätzlichen Eigenschaften.

Dank der Vererbung liess sich der Konflikt auflösen. Die Bahnen werden im Ilistal als IhB-Bergbahn erfasst, inklusive aller Zusätze. Nachdem IhBBergbahn eine Unterklasse von Bergbahn (gemäss Nationalverband) ist, kann jede IhBBergbahn auch als normale Bergbahn gelesen werden. Daher können die Ilistaler ihre Daten unverändert dem nationalen Verband schicken.

- Vererbung kann auch dazu benutzt werden, föderalistische Eigenheiten zu unterstützen.

Genau genommen liegt es am Polymorphismus, der durch die Vererbung ermöglicht wird: Jedes Exemplar einer Unterklasse kann immer auch als zur Oberklasse gehörend betrachtet werden (vgl. Abschnitt 5.5). Damit kann der nationale Verband die Daten von jeder Bergbahn im Land verarbeiten, auch wenn es eigentlich sich um ein Exemplar einer örtlichen Unterklasse von «Bergbahn» handelt, die der nationale Verband gar nicht kennt.

Die Vererbung geht bei INTERLIS sehr weit: Nicht nur Klassen und Themen, sondern auch Wertebereiche (Typen), Sichten, Grafikdefinitionen, in einem gewissen Sinn sogar Einheiten können übernommen und präzisiert werden.

5.7 Einfach- und Mehrfachvererbung

Einige Modellierungssprachen erlauben es, dass von mehreren Grund-Elementen gleichzeitig geerbt wird. Eine Klasse kann so mehrere Oberklassen gleichzeitig verfeinern.

Es ist in der Informatik umstritten, wie zweckmässig dies ist. Modelle mit Mehrfachvererbung sind häufig weniger übersichtlich. INTERLIS kennt daher nur die Einfachvererbung.

6. Das Datenmodell unter der Lupe

6.1 Taler und Groschen – Numerische Datentypen

6.1.1 Wertebereich

Ein Einzelbillett für eine einfache Fahrt mit der Standseilbahn auf das Ilshorn kostet 10 Ahländer Taler, ein Jahressportpass 635 Taler. Wie soll nun der Preis eines Billetes modelliert werden?

Wer sich an Programmiersprachen gewohnt ist, denkt sicher sofort an ganze Zahlen («Integer») oder Fließkommazahlen («Real»). Bei vielen Programmiersprachen und Datenbanken lässt sich festlegen, wie viel Speicherplatz eine Zahl benötigt; hieraus folgen Grösse und Genauigkeit der speicherbaren Zahlen («short integer», «long integer», «double precision», etc.). Man überlegt sich dann, in welchem Bereich die Werte etwa liegen können, und wählt die passende Speicherform.

Bei der Modellierung auf konzeptueller Stufe möchte man sich aber eigentlich nicht um die Speicherform kümmern. Es ist aber möglich, den höchsten und den tiefsten zulässigen Wert anzugeben und dabei gleich noch die Stellenzahl sowie den Exponenten festzulegen.

Da sich die Preise zwischen 10 und 635 Talern bewegen, könnte man also schreiben:

```
Preis: 10 .. 635;
```

Damit würde man aber auch festlegen, dass ein Preis nie kleiner als 10 Taler und nie grösser als 635 Taler sein kann. Das möchte man natürlich nicht. Als untere Schranke ist 0 plausibel. Aber die obere Schranke? Mehr als 10'000 Taler werden es ja kaum sein. Also 10'000? Wo eine Schranke auf Grund der Anwendung nicht genau festgelegt werden kann, macht es jedenfalls keinen Sinn, den Wert so zu wählen, dass eine kürzere Speicherform (z.B. *short integer* mit Werten zwischen -32768 und 32767) gerade nicht mehr ausreicht.

Dabei ist es natürlich auch wichtig zu überlegen, wie viele Nachkommastellen nötig sind. Wenn die Preise auch Groschen aufweisen können, werden die Schranken eben mit jeweils zwei Nachkommastellen festgelegt. Bei einem Attribut, das z.B. ein Investitionsbudget darstellt, möchte man sich nur mit Millionen abgeben; wo es nur um Grössenordnungen geht, reichen zwei Ziffern. Diese können sich aber auf Tausend, eine Million oder gar eine Milliarde beziehen.

```

Preis: 0.00 .. 9999.99;
Budget: 0.00E6 .. 999.99E6;
Bruttosozialprodukt: 0.0E0 .. 9.9E20;

```

Die Anzahl der Stellen bezeichnet die Genauigkeit: «4.3 Millionen» ($4.3E6 = 4.3 \cdot 10^6$) ist eine weniger genaue Angabe als «4300000».

6.1.2 Einheiten

Sind das jetzt Taler, Franken, Euro oder Dollar? Die Einheit ist – nicht nur beim Geld – von entscheidender Bedeutung. Darum ist es empfehlenswert, die Einheit nicht nur als Kommentar oder als Teil des Attributnamens anzugeben, sondern als Bestandteil des Typs:

```

UNIT
  Taler EXTENDS MONEY;

CLASS Billettart =
  Preis: 0 .. 9999 [Taler];
END Billettart;

```

Taler (oder Schweizerfranken, Euro, Dollar, etc.) werden als Einheit definiert und können dann bei der Definition eines numerischen Typs benutzt werden. Die meisten gängigen Einheiten (darunter auch CHF, EUR, USD) müssen jedoch nicht selbst definiert werden, sondern stehen über das spezielle Einheiten-Modell (units.ili, vgl. Abschnitt 3.3) zur Verfügung.



Im Interesse der Klarheit wird empfohlen, immer Einheiten anzugeben. Das INTERLIS-Einheitenmodell umfasst neben zahlreichen physikalischen Einheiten auch solche für verbreitete Währungen und für Anzahl, Prozente und Promille.

6.1.3 Numerische Typen erben

Da die Ilistalbahnen nie Billette abgeben werden, die mehr als tausend Taler kosten, könnte man auf die Idee kommen, dies in einer eigenen Klasse zu definieren:

```

CLASS Billettart (EXTENDED) =
  Preis (EXTENDED): 0 .. 1000 [Taler];
END Billettart;

```

In dieser Anwendung macht dieses konkrete Beispiel eher wenig Sinn. Es gibt aber durchaus Fälle in der Praxis, wo es sinnvoll ist, den ererbten Wertebereich einzuschränken.

Die Ilistaler dürfen nun allerdings nicht die Wertebereiche des nationalen Modells beliebig verändern: Es gilt das Prinzip, dass jeder Ilistaler Wert auch im Basismodell des Nationalverbands zulässig sein muss. Sonst wäre nicht mehr sichergestellt, dass der nationale Verband problemlos Daten aus dem Ilistal übernehmen kann.

Eine Preisangabe $0 \dots 1000$ ist also eine zulässige Einschränkung des national vorgesehenen Bereichs $0 \dots 9999$. Würde man den Wertebereich im Ilistal aber ausdehnen – etwa mit der Angabe $0 \dots 15000$ – dann wären Billettarten, die zwischen 10000 und 15000 Taler kosten, aus der Sicht des nationalen Verbandes nicht korrekt. Eine solche Definition ist darum unzulässig.

Die Ilistaler wollen auch die Möglichkeit haben, Billette auszugeben, deren Preis nicht zwingend ein voller Talerbetrag ist. Man möchte folgende Definition schreiben:

```
CLASS Billettart (EXTENDED) =
  Preis (EXTENDED): 0.00 .. 1000.00 [Taler];
END Billettart;
```

Aber ist das mit dem Basismodell verträglich? Ja, denn jeder Wert gemäss Ilistaler Modell (z.B. 7.35) kann mit mathematischer Rundung auf einen korrekten Wert des nationalen Modells abgebildet werden (z.B. 7).

Die Definition wäre dann unzulässig, wenn sie Werte umfassen würde, die gerundet das Basismodell verletzen. Beispielsweise würde ein Maximalwert von 9999.99 nach dem Runden zu 10000 – dies wäre mehr als die vom nationalen Verband vorgegebenen 9999. Dagegen könnten die Ilistaler einen Bereich 0.00 .. 9999.49 definieren, ohne dem Basismodell zu widersprechen, denn 9999.49 ergibt nach der Rundung wieder 9999.

Ebenfalls unzulässig ist es, im spezialisierten Modell auf Genauigkeit zu verzichten. Würde das nationale Tourismusmodell zum Beispiel einen Bereich von 0.00 .. 1000.00 vorsehen, könnten die Ilistaler in ihrer Spezialisierung nicht die Angabe 0 .. 1000 machen.

Und noch etwas: Die Einheiten der Erweiterung müssen immer mit derjenigen der Basis übereinstimmen!

6.1.4 Schranken noch unbekannt

Kann der nationale Verband festlegen, welcher Preis für ein Billett zulässig ist? Sind die Schranken noch völlig unbekannt, kann – im Rahmen abstrakter Attribute – auf ihre Angabe verzichtet werden. Es ist aber möglich, dennoch bereits eine Einheit festzulegen.

```
Preis (ABSTRACT): NUMERIC [Ahland.Taler];
```

Vielleicht ist sogar die Einheit noch unklar. Man weiss erst, dass es sich z.B. um eine Währung handelt.

```
Preis (ABSTRACT): NUMERIC [MONEY];
```

Damit ist immerhin der Charakter der Einheit festgelegt. In einer Erweiterung können damit nur noch Einheiten definiert werden, die Konkretisierungen der abstrakten Einheit MONEY sind (zu abstrakten Eigenschaften siehe auch Abschnitt 5.4).

```
Preis (EXTENDED): 0 .. 10000 [CHF]; !! erlaubt
Preis (EXTENDED): 0 .. 2000 [USD]; !! erlaubt
Preis (EXTENDED): 0 .. 1000 [m];    !! verboten, da Meter eine Konkre-
                                     !! tisierung von LENGTH und nicht
                                     !! von MONEY ist.
```

6.2 Arten von Bergbahnen – Modellierung von Arten von Objekten

Für eine grobe Übersicht genügt es, die Bergbahnen grob einzuteilen: Zahnradbahn, Standseilbahn, Luftseilbahn, Skilift, Sessellift, Gondelbahn. Im einfachsten Fall wird die Art als Textattribut festgehalten.

```

CLASS Bergbahn =
  Name: TEXT*100;
  Art: TEXT*50;
END Bergbahn;

```

Als Folge ist die Person, welche die Daten erfasst, in der Beschreibung sehr frei. Seilbahn, Schwebebahn, Skilift, Ski-Lift – es ist zu befürchten, dass ein rechter Wildwuchs an Bezeichnungen entsteht. Vermeiden lässt sich dies mit einer Aufzählung.

```

CLASS Bergbahn =
  Name: TEXT*100;
  Art: (Zahnradbahn,
        Standseilbahn,
        Luftseilbahn,
        Skilift,
        Sessellift,
        Gondelbahn);
END Bergbahn;

```

Da jetzt alle zulässigen Möglichkeiten aufgezählt sind, herrscht Ordnung. Oft möchte man nun noch weitere Attribute anfügen, z.B. die Anzahl der Plätze in der Bahn. Bei Stand- und Luftseilbahn ist dies das Fassungsvermögen der ganzen Kabine, bei den Ski- und Sesselliften die Anzahl Personen pro Einzelfahrt. Bei der Zahnradbahn, wo mehrere Wagen zusammengekuppelt werden können, macht die Angabe jedoch keinen Sinn. Dafür interessiert dort vielleicht das Zahnstangensystem. Soll jetzt die Klasse Bergbahn einfach alle Attribute aufweisen, die zum Beschreiben der verschiedenen Arten nötig sind?

Wenn die verschiedenen Arten jeweils eigene Eigenschaften (Attribute oder Beziehungen) aufweisen, ist es sinnvoll, eigene Klassen zu definieren, welche die Basisklasse beerben (vgl. Kapitel 5).

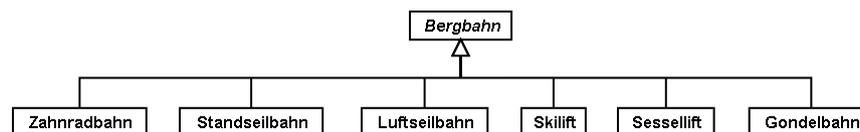


Abbildung 30: Zahnradbahnen, Standseilbahnen, etc. sind spezielle Bergbahnen. Es gibt jedoch keine Bergbahnen an und für sich: Alle «konkreten» Bergbahnen gehören immer zu einer der Unterklassen. *Bergbahn* ist damit eine abstrakte Klasse, was im Diagramm mittels Schrägschrift bezeichnet wird.

Es gibt aber keine Bergbahnen, die ausschliesslich Bergbahn sind und nicht gleichzeitig auch einer Unterklasse angehören. Die Klasse Bergbahn wird dann als «abstrakt» deklariert. Eine konkrete Bergbahn muss dann immer eine Zahnradbahn, eine Luftseilbahn, usw. sein.

In der textuellen Schreibweise von INTERLIS 2 werden abstrakte Klassen mit der Angabe (ABSTRACT) in Klammern bezeichnet. Nur nebenbei: Das INTERLIS-Einheitenmodell «Units» kennt eine Einheit «CountedObjects» für abgezählte Objekte wie zum Beispiel die Zahl der Personen in einer Luftseilbahnkabine.

```
CLASS Bergbahn (ABSTRACT) =
  Name: Text * 100;
END Bergbahn;

CLASS Zahnradbahn EXTENDS Bergbahn =
  Stangensystem: (Riggenbach, Abt, vonRoll);
END Zahnradbahn;

CLASS Standseilbahn EXTENDS Bergbahn =
  Fassungsvermoegen: 0 .. 999 [Units.CountedObjects];
END Standseilbahn;

CLASS Luftseilbahn EXTENDS Bergbahn =
  Fassungsvermoegen: 0 .. 999 [Units.CountedObjects];
END Luftseilbahn;

CLASS Skilift EXTENDS Bergbahn =
  PersonenProFahrt: 0 .. 10 [Units.CountedObjects];
END Skilift;

CLASS Sessellift EXTENDS Bergbahn =
  PersonenProFahrt: 0 .. 24 [Units.CountedObjects];
END Sessellift;

CLASS Gondelbahn EXTENDS Bergbahn =
  Fassungsvermoegen: 0 .. 99 [Units.CountedObjects];
END Gondelbahn;
```

Für die Sitzung wurde extra ein Eisenbahner eingeladen, der längere Zeit über Zahnradbahnen referierte. Die Anwesenden lernten viel darüber, was für Zahnstangensysteme weltweit im Einsatz sind und welche Vor- und Nachteile sie jeweils besitzen. Schliesslich fragten sich die Ilistaler aber, was genau die Zahnstangensysteme eigentlich mit ihrem Projekt zu schaffen haben. Es konnte sich auch niemand vorstellen, wie diese und andere Angaben jemals in einem zukünftigen Ausbauschnitt von Belang werden könnten. Daher wurde dieses Modell verworfen, weil es zu sehr ins Detail geht und am Ende lediglich Kosten für das Erfassen und Pflegen unnötiger Daten verursacht hätte.

Siehe auch Abschnitt 5.1 zur Verlockung, beim Modellieren allzu sehr ins Detail zu gehen.

6.3 Gibt es auch hellblaue Skipisten? – Strukturierte Aufzählungen

6.3.1 Gewöhnliche Aufzählungen und ihr Erbrecht

Um den Schwierigkeitsgrad von Skipisten grob zu beschreiben, wurden drei Farben gewählt: blau, rot, schwarz. Es soll nur diese und keine anderen Schwierigkeitsgrade geben. Zudem sind sie geordnet. Blau bezeichnet eine einfache Piste, eine rote Piste ist schwieriger als eine blaue, eine schwarze am anspruchsvollsten. Dies wird mit der folgenden Definition beschrieben:

```
CLASS Piste =
  Schwierigkeitsgrad: (blau, rot, schwarz: FINAL) ORDERED;
END Piste;
```

Würde die FINAL-Angabe fehlen, könnte die Aufzählung in einer Erweiterung noch ergänzt werden. Zum Beispiel könnte dies bei der Art von Bergbahnen Sinn machen:

```

!! Modell des Nationalen Tourismusverbandes
CLASS Bergbahn =
  Art: (Zahnradbahn, Standseilbahn, Luftseilbahn,
       Skilift, Sessellift, Gondelbahn);
END Bergbahn;

!! Modell Ilistal
CLASS IhBBergbahn EXTENDS Bergbahn =
  Art (EXTENDED): (Schneebus);
END IhBBergbahn;

```

In der erweiterten Klasse wird der Aufzählung noch das Element Schneebus – das neuste vom Neuen – am Ende der bisherigen Aufzählung beigefügt. Aber was fängt der nationale Tourismusverband damit an? Dort ist «Schneebus» doch ein unbekannter Wert.

- ▶ Jede (horizontale) Erweiterung kann durch weitere Werte ergänzt werden, solange dies nicht ausdrücklich mit **FINAL** ausgeschlossen wird. Interessiert sich jemand für die Werte nur allgemein gemäss der Basisklasse, werden diese Werte alle in den Wert **OTHER** übersetzt.

Für die Basisklasse ist der Wert Schneebus (und allfällige weitere Werte) nur noch als OTHER erkennbar. Wird jedoch FINAL angegeben, kann der Wert OTHER nicht mehr auftreten. Ist eine Aufzählung zyklisch (**CIRCULAR**) definiert, sind solche Ergänzungen nie möglich, heisst doch zyklisch, dass nach dem höchsten Wert wieder der niedrigste kommt und man sonst gar nicht wüsste, welches der höchste ist.

```
Windrichtung: (N, NE, E, SE, S, SW, W, NW) CIRCULAR;
```

6.3.2 Unteraufzählungen

Man hatte also beschlossen, die verschiedenen Arten von Bergbahnen nicht mit einer ganzen Landschaft von Klassen zu modellieren. Aber die Bahnfreunde waren nicht recht einverstanden: Das Stangensystem der Zahnradbahnen könnte ja vielleicht doch noch irgendwann einmal von Interesse sein...

Für jeden Wert einer Aufzählung kann eine Unteraufzählung definiert werden. Dies kann direkt innerhalb der Basisdefinition oder erst in einer Erweiterung erfolgen.

```

CLASS IhBBergbahn EXTENDS Bergbahn =
  Art (EXTENDED): (Zahnradbahn (Riggenbach, Abt, vonRoll));
END IhBBergbahn;

Wochentag: (Werktag (Montag, Dienstag, Mittwoch,
                   Donnerstag, Freitag, Samstag),
           Sonntag);

```

Wird eine solche Unteraufzählung in einer Erweiterung definiert, ist sie aus der Sicht der Basis einfach nicht von Belang. Aus der Sicht des nationalen Tourismusverbands würde also auch eine Riggenbach-Zahnradbahn eine Zahnradbahn sein.

Auch Unteraufzählungen können wieder um weitere Werte ergänzt werden, sofern ihr letzter Wert nicht als FINAL erklärt wurde. Die einzelnen Werte einer Unteraufzählung können

zudem wiederum durch Unteraufzählungen präzisiert werden, so dass ganze Aufzählungsbäume entstehen.

6.4 Ilistaler halten sich kurz – Zeichenketten und ihre Erbregein

Bezeichnungen können grundsätzlich beliebig lange Namen enthalten. Der nationale Verband hat jedoch festgelegt, dass der Name einer Bergbahn höchstens 100 Zeichen aufweisen darf. In der Regel sind die Namen natürlich durchaus kürzer, man wollte einfach sicher gehen.

```
STRUCTURE Bahnbezeichnung EXTENDS Bezeichnung =
  Name (EXTENDED): TEXT*100;
END Bahnbezeichnung;
```

Ist die Länge eines Text-Attributes beliebig oder noch vollkommen unbekannt, kann auf die Angabe der Länge verzichtet werden. Ist es klar, dass die Länge im Rahmen einer Klassenerweiterung noch festgelegt wird, wird das Attribut als abstrakt bezeichnet:

```
Beschreibung (ABSTRACT): TEXT;
```

Manche Bahnen im Ilistal haben eine Web-Kamera installiert, die laufend die Umgebung der Bergstation aufnimmt. Interessierte Touristen können so sehen, ob sich die Reise lohnt. Die Internet-Adresse des aktuellen Bildes ist ebenfalls eine (etwas besondere) Art von Text.

```
CLASS IhBBergbahn =
  ...
  BildBergstation: URI;
  ...
END IhBBergbahn;
```

Internet-Adressen haben aber nichts mit einem Schweizer Kanton zu tun – oder wenn schon, dann allenfalls mit Genf, wo am CERN der erste Web-Browser entwickelt wurde. URI ist schlicht die Abkürzung von *Uniform Resource Identifier*. Die meist für Web-Seiten benutzten *Uniform Resource Locators (URLs)* sind spezielle URIs.

6.5 Windstille – Fakultative und obligatorische Attribute

Als aktuelle Betriebsdaten werden auch aktuelle Wetterdaten wie Temperatur sowie Richtung und Stärke des Windes gemeldet. Bei Windstille macht die Angabe der Windrichtung keinen Sinn. Die anderen Angaben sollen immer gemacht werden.

- ▶ Die Tatsache, dass ein Attribut **undefiniert** sein kann, bzw. dass es immer definiert sein muss, ist Teil des Modells.



Undefiniert ist nicht einfach 0 oder sonst irgendein etwas besonderer Wert. Es ist ein eigenständiger Wert, der genau die Tatsache der Undefiniertheit wiedergibt.

In INTERLIS 2 schreibt man zum Beispiel:

```
CLASS Wetter =  
  Temperatur: MANDATORY -50 .. 50 [oC];  
  Windrichtung: (N, NE, E, SE, S, SW, W, NW) CIRCULAR;  
  Windgeschwindigkeit: MANDATORY 0 .. 200 [kmh];  
END Wetter;
```

Temperatur und Windgeschwindigkeit sind also obligatorisch (MANDATORY). Da die Windrichtung nicht obligatorisch verlangt wird, ist sie fakultativ. Der konkrete Wert darf damit undefiniert sein. In Erweiterungen ist es zulässig, aus fakultativen Attributen obligatorische zu machen. Obligatorische Attribute dürfen aber nicht zu fakultativen werden, da gemäss der Basisklasse der Wert «undefiniert» nicht erlaubt ist.

6.6 Wartezeiten und Fahrzeiten – Wertebereiche

Wartezeiten an den Bergbahnen und die Fahrzeiten der Bergbahnen werden beide in Minuten festgehalten.

```
CLASS Bergbahn =  
  Fahrzeit: 0 .. 200 [min];  
END Bergbahn;
```

```
CLASS Bergbahnstatus =  
  Wartezeit: 0 .. 200 [min];  
END Bergbahnstatus;
```

Beide Eigenschaften können Werte aus demselben Bereich annehmen. Mit einem ausdrücklich definierten Wertebereich (DOMAIN) lässt sich diese Gemeinsamkeit betonen:

```
DOMAIN  
  ZeitdauerInMinuten = 0 .. 200 [min];
```

```
CLASS Bergbahn =  
  Fahrzeit: ZeitdauerInMinuten;  
END Bergbahn;
```

```
CLASS Bergbahnstatus =  
  Wartezeit: ZeitdauerInMinuten;  
END Bergbahnstatus;
```

6.7 Wo liegt das llistal? – Koordinatentypen

6.7.1 Grundsätzliches zu Koordinatentypen

Mit der Frage «Wo?» ist die Vorstellung eines punktförmigen Ortes in der realen Welt verbunden. Ein solcher Ort kann mittels einer Koordinate beschrieben werden. Eine Koordinate ist typischerweise ein Zahlenpaar das die Lage, oder ein Zahlentripel das die Lage und Höhe, eines Ortes beschreibt.

Für jede Dimension eines Koordinatentyps muss darum wie für jeden numerischen Typ festgelegt werden, in welchem Zahlenbereich die zulässigen Werte liegen dürfen und welche Einheit mit ihr verbunden ist.

```
Lage: COORD 500.00 .. 91000.00 [m],
      700.00 .. 23000.00 [m];
```

```
XLage: 500.00 .. 91000.00 [m];
YLage: 700.00 .. 23000.00 [m];
```

Der Unterschied zwischen einem Lageattribut mit einem Koordinatentyp und je einem numerischen Attribut für die X- und die Y-Richtung ist auf den ersten Blick klein. Dank der Definition als Koordinatentyp ist es aber offensichtlich, dass die beiden Angaben zusammengehören. Diese Eigenschaft kann durch die Programmpakete auch ausgenutzt werden. So sind viele Programme dafür eingerichtet, kartesische Koordinatenwerte grafisch darzustellen.

Kartesische Koordinatenwerte? Als kartesische Koordinaten werden Koordinaten bezeichnet, deren Dimensionen senkrecht aufeinander stehen. Mit der Definition der obigen Lagekoordinaten wird also ein rechteckiges Fenster von etwa 90 mal 22 Kilometer Ausdehnung beschrieben. Ein Rückfall ins Mittelalter? Ist nun die Erde im Ilistal wieder zur Scheibe geworden?

6.7.2 Die umwickelte Zwetschge – Was ist ein Koordinatensystem?

Schon für Ptolemäus war die Erde eine Kugel. Die Vermesser (oder die Geodäten, wie sie heissen, wenn es um die gehobeneren Fragen der Vermessungstechnik geht) mussten sich schon lange von dieser Sicht abwenden, weil sie allzu sehr vereinfacht.

Eine brauchbare Annäherung der Erdoberfläche ist das Ellipsoid, also jene Fläche, die entsteht, wenn sich eine Ellipse um ihre zentrale Achse dreht.

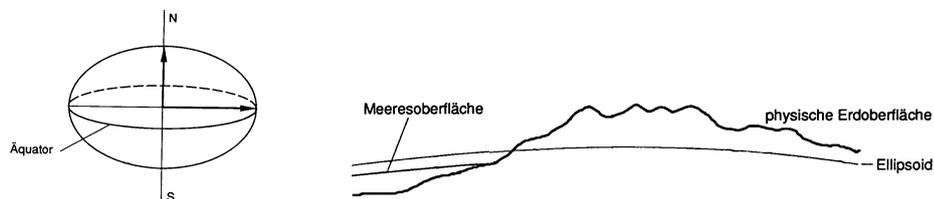


Abbildung 31: Dreht sich eine Ellipse um ihre eigene Achse, entsteht im Raum eine flachgedrückte Kugel. Mit einem solchen Ellipsoid kann die Form der Erdoberfläche angenähert werden. (Alle Abbildungen in diesem Abschnitt und in Abschnitt 6.7.5 aus: K. Christoph Graf, Verwendung geodätischer Abbildungen bei der Geocodierung von Satelliten-Bildern. Zürich, 1988. Teilweise wurden die Illustrationen vereinfacht. Ursprüngliche Bildquellen siehe dort).

Je nach Weltgegend werden anders gelegene Ellipsoide benutzt, sonst würde die Annäherung zu ungenau. Beispielsweise verwendet die Schweiz das gleiche Ellipsoid wie Deutschland, aber ein leicht anderes als Schweden oder Frankreich.

Als räumliche Gebilde sind Ellipsoide jedoch etwas mühselig zu handhaben. Aus diesem Grund bilden Geodäten das Ellipsoid auf eine Fläche ab. Hierzu legen sie einen Zylinder oder Kegel an das Ellipsoid an und beleuchten es von innen, so dass das Landschaftsbild auf den Zylinder oder den Kegel projiziert wird.

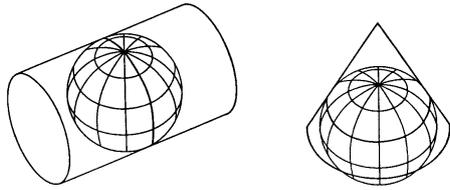


Abbildung 32: Das Ellipsoid wird in einen Zylinder (links) oder Kegel (rechts) gewickelt. Anschliessend wird es von innen beleuchtet.

Als nächstes wird der Zylinder oder Kegel mit einer Schere aufgeschnitten, abgerollt und flach auf den Tisch gelegt – fertig ist die Karte!

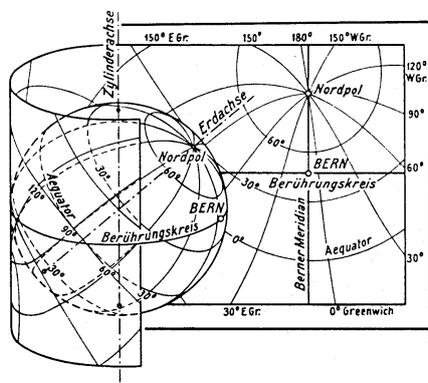


Abbildung 33: Nach erfolgter Projektion wird der Zylinder (bzw. Kegel) aufgeschnitten und abgerollt. Ein gewölbter Körper wie etwa ein Ellipsoid oder eine Kugel könnte zwar aufgeschnitten, aber nicht flach abgerollt werden.

Zuletzt werden feine, rechtwinklig aufeinander stehende Linien über die Karte gelegt: Das **Koordinatensystem** der Karte. Bei jedem Koordinatentyp muss darum auch festgelegt werden, welches Koordinatensystem ihm zu Grunde liegt.

```
Lage: COORD 480000 .. 850000.00 [m] {AhlandSys[1]},
      60000 .. 320000.00 [m] {AhlandSys[2]};
```

Die erste Dimension der Koordinate entspricht der ersten Achse des Koordinatensystems mit Namen «AhlandSys», die zweite Dimension der zweiten Achse des gleichen Systems.

6.7.3 Angaben zum Koordinatensystem – Metadaten

Ist «AhlandSys» ein kartesisches, ein ellipsoidisches System? Wie heissen die Achsen? Gibt es Zusammenhänge (z.B. Kartenprojektionen) zu anderen Koordinatensystemen? All diese Angaben können selbst wieder mittels Daten beschrieben werden. Damit klar ist, wie diese Daten strukturiert sind, wird dafür ebenfalls ein Datenmodell formuliert. Ein solches Modell heisst Metamodell, die zugehörigen Daten Metadaten, weil sie dazu dienen, die eigentlichen Daten zu beschreiben.

Die Daten zu einem Metamodell sind in einem anderen, formaleren Sinn «meta» als Angaben zu Herkunft oder Preis (vgl. Abschnitt 3.3). Für beides ist jedoch unglücklicherweise dieselbe Bezeichnung verbreitet.

In den einfachen Fällen, wo es auf Grund des Anwendungs- und Einsatzgebietes eines Datenmodells klar ist, zu welchem Koordinatensystem die Koordinaten gehören, kann auf die explizite Angabe des Koordinatensystems verzichtet werden. Es macht aber Sinn, das Koordinatensystem mindestens im Namen des Koordinatentyps anklingen zu lassen.

```
LandesKoord = COORD 500.00 .. 91000.00 [m],
              700.00 .. 23000.00 [m];
```

```
Lage: LandesKoord;
```

Um Verwechslungen auszuschliessen haben die Ilistaler eine präzise Definition vorgezogen:

```
REFSYSTEM BASKET CoordSystems ~ CoordSys.CoordsysTopic
OBJECTS OF GeoCartesian2D: AhlandSys;
```

Sie haben auf der Grundlage des allgemeinen Modells für Koordinatensysteme (CoordSys) ihr Landessystem präzise definiert. Für die Lage wurde dafür in den entsprechenden Daten ein Objekt der Klasse GeoCartesian2D mit dem Namen AhlandSys eingetragen. Die Existenz dieses Dateneintrags wird mittels OBJECTS OF im Modell angemerkt. Das Koordinatensystem „AhlandSys“ ist damit im Modell verfügbar. Bei der Anwendung des Systems muss der Name des Metadatenbestands (CoordSystems) nur erwähnt werden, wenn im aktuellen Modellierungsteil mehrere solche Metadatenbestände definiert sind.

```
LandesKoord = COORD 500.00 .. 91000.00 [m] {CoordSystems.AhlandSys[1]},
              700.00 .. 23000.00 [m] {CoordSystems.AhlandSys[2]};
```

6.7.4 Verschiedene Koordinatensysteme

Damit denjenigen Touristen, die über einen einfachen GPS-Empfänger verfügen, ein besonderer Service geboten werden kann, möchten die Ilistaler ihre Koordinaten auch als geografische Koordinaten im globalen WGS84-System anbieten.

```
WGS84Coord = COORD -90.00000 .. 90.00000 [Angle_Degree] {WGS84[1]},
              0.00000 .. 359.99999 CIRCULAR [Angle_Degree]
              {WGS84[2]};
```

```
CLASS Bergbahn =
  LageTalstation: LandesKoord;
  LageTalstationWGS: WGS84Koord;
  ....
END Bergbahn;
```

Nun ist es aber offensichtlich, dass die beiden Attribute einen direkten Zusammenhang haben. Landeskoordinaten können doch in WGS84-Koordinaten umgerechnet werden. Die detaillierte Definition einer solchen Umrechnung ist aber nicht Aufgabe einer konzeptuellen Beschreibung der Daten. Es ist aber wünschenswert anzugeben, dass die einen Koordinaten aus den anderen gerechnet werden können.

```

!! Umrechnung von Koordinaten im Ahländer Landessystem zu WGS84.
!! Funktionen werden in Abschnitt 7.2 diskutiert.
FUNCTION AhlandToWGS84 (Ah: Ahland.LandesKoord): WGS84Koord;

CLASS Bergbahn =
  LageTalstation: Ahland.LandesKoord;
  LageTalstationWGS: WGS84Koord := AhlandToWGS84 (LageTalstation);
  ....
END Bergbahn;

```

6.7.5 Dreidimensionale Koordinaten

Den Skifahrern und Wanderern rund um das Ilishorn genügen natürlich die Lagekoordinaten nicht. Grosse Höhendifferenzen lassen die Skifahrerherzen höher schlagen, während der Wanderer Schweissperlen oder schlotternde Knie befürchten muss. Höhen sind gefragt! Koordinatentypen können darum auch drei Dimensionen aufweisen.

```

LandesKoord3 = COORD 500.00 .. 91000.00 [m] {AhlandSys[1]},
                700.00 .. 23000.00 [m] {AhlandSys[2]},
                0.00 .. 9000.00 [m] {AhlandHoechenSys[1]};

WGS84Koord = COORD -90.00000 .. 90.00000 [Angle_Degree] {WGS84[1]},
                0.00000 .. 359.99999 CIRCULAR [Angle_Degree]
                {WGS84[2]},
                -2000.00 .. 9000.00 [m] {WGS84H[1]};

```

Bei den Höhen stellt sich noch ein besonderes Problem. Wo ist eigentlich die Höhe 0? Wie kann man die Höhe eines Punktes gegenüber dieser Höhe 0 bestimmen? Die Geodäten unterscheiden vor allem zwischen den Höhen gemäss dem Schwerefeld der Erde (Schwere- oder Geoid-Höhe; 0 ist die Höhe der gedachten Fortsetzung des Meeres unter den Kontinenten) und Höhen gemäss der geometrischen Annäherung der Erde (Ellipsoid-Höhe; 0 ist die Oberfläche des Ellipsoids).

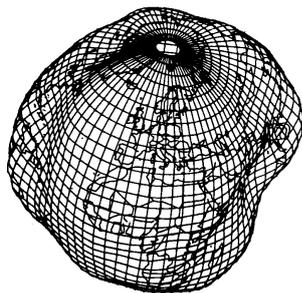


Abbildung 34: Das Schwerefeld der Erde: Beim Geoid wird die Meeresoberfläche in Gedanken unter den Kontinenten fortgesetzt. Gebirgsmassive, Meeresgräben etc. beeinflussen das Schwerefeld und verformen so die gedachte Wasseroberfläche. Diese Zeichnung ist sehr stark überhöht.

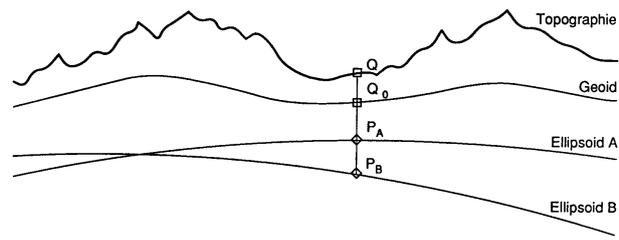


Abbildung 35: Je nach gewähltem Bezugssystem besitzt Punkt Q eine andere Höhe.

Die Landeskoordinatensysteme verwenden typischerweise Geoidhöhen. Darum bezieht sich die dritte Dimension der Landeskoordinaten auch nicht einfach auf die dritte Achse des Landessystems, sondern auf die erste Achse eines speziellen Höhensystems.

Dagegen werden die Koordinaten bei GPS-Messungen rein geometrisch aus Satellitenpositionen bestimmt, ohne dass das Schwerefeld der Erde eine Rolle spielen würde. WGS84-Höhen sind also Ellipsoidhöhen.

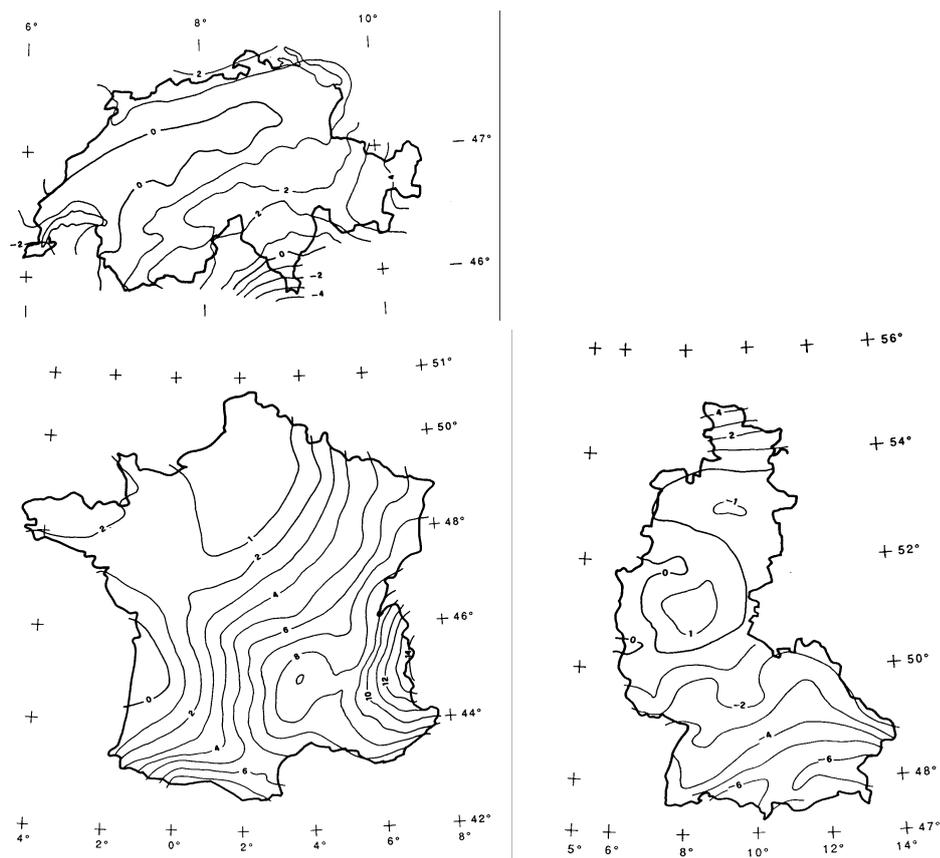


Abbildung 36: Die Schwerehöhe kann bis zu einigen Metern von der Ellipsoidhöhe abweichen. Gezeigt sind die Abweichungen zum jeweils üblichen Ellipsoid der Schweiz, von Frankreich und dem ehemaligen West-Deutschland.

Die Umrechnung zwischen Schwerehöhen und Ellipsoidhöhen kann vor allem dort ein Problem sein, wo der Bereich der zulässigen Koordinaten ein Gebiet abdeckt, dessen Schwerefeld nicht mehr homogen ist. Zum Glück sind diese Fragen bei der Modellierung nur von geringer Bedeutung. Einen kleinen Gedanken sind sie aber dennoch wert.

6.8 Ist Null im Norden? – Festlegungen für Winkel und Richtungen

Wie gross ist ein rechter Winkel? 90 Grad oder $\pi / 2$? Das ist eine Frage der Einheit, die verwendet wird. Aber wann gilt der Winkel als positiv, wann als negativ? Zu einem Winkeltyp gehört darum der Drehsinn: Uhrzeigersinn oder Gegenuhrzeigersinn.

DOMAIN

```
WinkelImUhrzeigersinn = -179 .. 180 CIRCULAR CLOCKWISE;
WinkelImGegenuhrzeigersinn = -179 .. 180 CIRCULAR COUNTERCLOCKWISE;
```

Wenn wir auf dem Ilishorn stehen, möchten wir vielleicht wissen, in welche Richtung wir schauen müssen, um das Krummhorn zu sehen. 50 Grad? 40 Grad? 310 Grad?

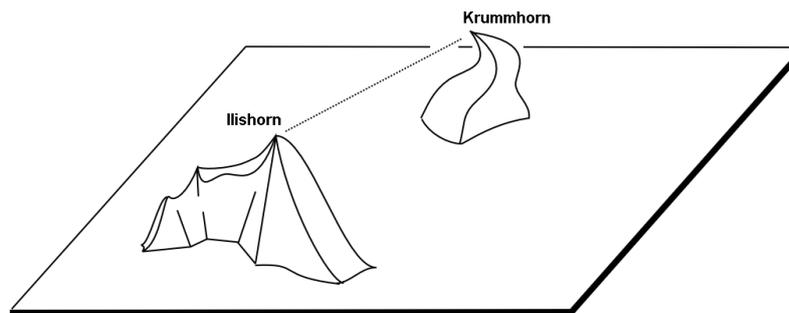


Abbildung 37: Wer das Ilishorn besteigt, wird für die Mühen mit einer prächtigen Bergsicht belohnt. Aber in welchem Winkel ist das Krummhorn zu sehen? Ist nicht klar, auf welches Koordinatensystem sich die Frage bezieht, lässt sich keine Antwort geben.

Es kommt eben drauf an, wo die Nullrichtung ist und wie die Richtungen drehen. Wenn von Richtungen gesprochen wird, muss darum immer auch von einem Bezugssystem gesprochen werden. Richtungen stehen darum in engem Zusammenhang mit Koordinatentypen. Es macht ja auch Sinn, die Distanz und die Richtung zwischen zwei über Koordinaten definierten Punkten zu bestimmen.

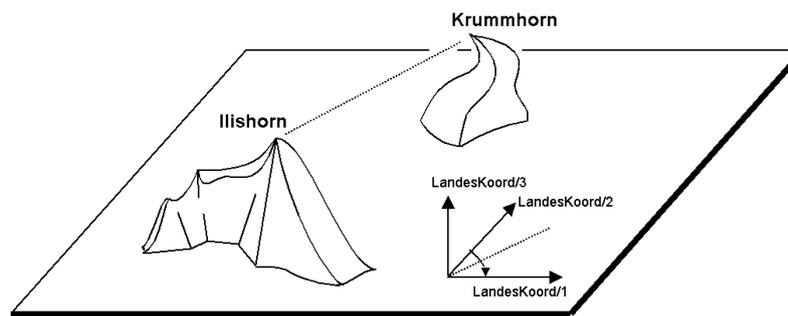


Abbildung 38: Die Angabe von Achsen und Drehsinn gehört zur Definition eines Koordinatensystems.

```
LandesKoord3 = COORD 500.00 .. 91000.00 [m] {AhlandSys[1]},
              700.00 .. 23000.00 [m] {AhlandSys[2]},
              -200.00 .. 14000.00 [m] {AhlandHoechenSys[1]},
              ROTATION 2 -> 1;
```

```
Richtung = 0.0 .. 359.9 CIRCULAR [Angle_Degree] {AhlandSys};
```

6.9 Ist eine Piste eine Linie oder eine Fläche? – Geometrietypen

6.9.1 Einfache konzeptuelle Sicht einer Linie

Vom Standpunkt der Skifahrer aus gesehen ist der Bedarf klar: Sie wollen wissen, wo die Piste beginnt, wo sie endet und wo sie grob durchführt. Gibt es ein Gasthaus am Pistenrand? Führt die Piste über freie Hänge oder durch den Wald? Für diese Information genügt es, den Pistenverlauf als Linie zu beschreiben.

Unter einem Linientyp darf man sich zunächst genau das vorstellen, was das Wort verspricht: Eine mehr oder minder komplizierte Verbindung zwischen zwei Punkten.

In diesem Sinn ist ein Linientyp nichts anderes als z.B. ein numerischer Typ oder besser noch ein Koordinatentyp. Da die an der Linie beteiligten Punkte durch Koordinaten beschrieben werden müssen, ist es zwingend, dass ein Linientyp immer mit einem Koordinatentyp verbunden sein muss.

In INTERLIS könnte man schreiben:

```
AhlandLinie = POLYLINE VERTEX Ahland.LandesKoord;

CLASS Piste =
  Verlauf: AhlandLinie;
END Piste;
```

Der Pistenverlauf wird mittels Linien beschrieben, die auf dem Ahländer Landes-Koordinatensystem basieren. Die Stützpunkte der Linien im Ahländer Landessystem stützen sich deshalb auf den Koordinatentyp des Landessystems ab.

6.9.2 Linienstücke

Es ist offensichtlich: Die Piste vom Ilishorn zur Ilisegg ist eine komplizierte Linie. Die Pisten bei den Ponyliften dagegen relativ einfach. Alles mit demselben Typ beschreibbar? Die Lösung liegt darin, dass die Linie als Ganzes in einzelne Linienstücke aufgeteilt wird. Jedes Linienstück ist selbst eine einfache Geometrie (z.B. eine Gerade, ein Stück eines Kreisbogens) und schliesst jeweils an das Vorgängerstück an.

Diesen Sachverhalt könnte man im konzeptuellen Modell auch darstellen. Das wäre aber eher eine überflüssige Belastung. Wenn man einmal weiss, dass Linien immer so aufgebaut sind, muss dies ja nicht mehr dargestellt werden.

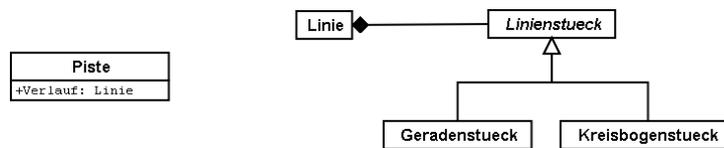


Abbildung 39: Der Verlauf einer Piste ist eine Linie. Diese bestehen ihrerseits aus einzelnen Linienstücken, von denen es verschiedene Arten gibt: Geradenstücke, Kreisbogenstücke, etc.

Es macht aber durchaus Sinn anzugeben, welche Arten von Linienstücken bei einem bestimmten Linientyp vorkommen dürfen.

```
AhlandLinie = POLYLINE WITH (STRAIGHTS, ARCS) VERTEX Ahland.LandesKoord;
```

Mit dieser INTERLIS 2-Definition wird angegeben, dass Linien dieses Typs Geraden- und Kreisbogenstücke aufweisen dürfen.

In vielen Fällen – so auch bei den Pisten – macht es keinen Sinn, dass eine Linie Schnittpunkte mit sich selbst aufweist. Solche Einschränkungen gehören auch zum konzeptuellen Modell. Aufgrund von Ungenauigkeiten beim Vermessen (und teils auch beim Berechnen) ist es jedoch möglich, dass eine an sich überlappungsfreie Form ganz leichte Überlappungen aufweist. Aus diesem Grund ist die maximal noch zulässige Überlappung Teil des Modells. Sie wird in den Einheiten der zugehörigen Koordinaten angegeben.

Nachdem das Ahländer Landes-Koordinatensystem Meter verwendet, sind mit dieser Definition Überlappungen bis zu 2 cm erlaubt:

```
AhlandLinie = POLYLINE WITH (STRAIGHTS, ARCS)
              VERTEX Ahland.LandesKoord
              WITHOUT OVERLAPS > 0.02;
```

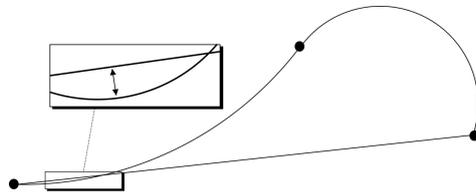


Abbildung 40: Kleine Überschneidungen sind manchmal nicht vermeidbar. Es ist Teil des Modells, wie gross die Überlappung (im Bild die Pfeilhöhe) höchstens sein darf.

6.9.3 Gerichtete Linien

Als Skifahrer erwartet man natürlich, dass die Linienstücke der Piste vom Ilshorn zur Ilisegg beim Ilshorn beginnen und bei der Ilisegg enden. Man möchte ja hinunterfahren und nicht die Steigfelle montieren! Zur Beschreibung anderer Objekte (z.B. der Wanderwege) ist jedoch die Richtung nicht von Bedeutung. Wo die Richtung der Linien von Bedeutung ist, soll dies im konzeptuellen Modell auch angegeben werden.

```
AhlandLinieGerichtet = DIRECTED POLYLINE VERTEX Ahland.LandesKoord;

CLASS Piste =
  Verlauf: AhlandLinieGerichtet;
END Piste;
```

6.9.4 Flächen

Für den Pistendienst der Ilshornbahnen stellte sich die Frage, ob die Beschreibung der Pisten für seine Zwecke genügt. Damit immer klar ist, welche Bereiche jeweils präpariert werden müssen, wird eine Darstellung als Fläche vorgezogen.

```
DOMAIN
  AhlandLinieGerichtet = DIRECTED POLYLINE WITH (STRAIGHTS, ARCS)
                        VERTEX Ahland.LandesKoord;

  AhlandFlaeche = SURFACE WITH (STRAIGHTS, ARCS)
                  VERTEX Ahland.LandesKoord;

CLASS Piste =
  Verlauf: AhlandLinieGerichtet;
  Praepariert: AhlandFlaeche;
END Piste;
```

Kurz vor der Ilisegg steht mitten in der Piste ein grosser Baum – oder anders gesagt, die Piste geht links und rechts am Baum vorbei.

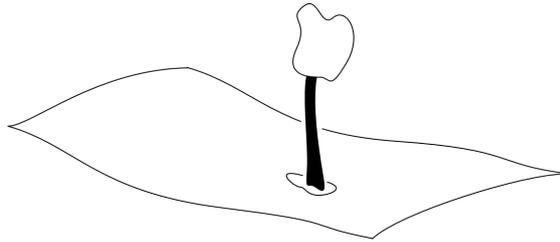


Abbildung 41: Mitten in der Piste steht ein grosser Baum. Für Skifahrer mag die Lage brenzlich sein, aber um das Datenmodell braucht man sich nicht zu sorgen: Trotz der Enklave ist die Piste eine einzige Fläche.

Ist die zu präparierende Fläche noch eine einzige Fläche? Mit Flächen – mindestens im Sinne von INTERLIS – sind immer zusammenhängende Bereiche gemeint. Auch wenn sie im Innern Aussparungen (Löcher, Enklaven) haben, sind es immer noch zusammenhängende Bereiche und können damit als eine Fläche beschrieben werden.

- ▶ Eine Fläche hat genau eine **äussere Begrenzung**. Sie darf keine, eine oder mehrere **innere Begrenzungen** (Enklaven) haben.

Oben beim Ilishorn liegen verschiedene Pisten anfänglich so nahe beieinander, dass eine gemeinsame präparierte Fläche entsteht. Welcher Flächenteil soll nun welcher Piste zugeordnet werden? Im Ilistäli kreuzen sich zwei Pisten. Damit wird die Fläche ja doppelt erfasst. Für die Abschätzung des Arbeitsaufwandes für die Präparierung stört das natürlich.

Der Pistendienst hat sich darum für eine andere Modellierung entschieden: Die zu präparierenden Flächen werden nicht direkt den Pisten zugeordnet, sondern als eigenständige Pistenabschnitte geführt. Jeder Pistenabschnitt ist eine Fläche. Die Pistenabschnitte sollen sich aber nie überlappen, da ein bestimmter Geländeabschnitt schliesslich nur einmal präpariert werden muss.

```
DOMAIN
  AhlandGebietseinteilung = AREA WITH (STRAIGHTS, ARCS)
                                VERTEX Ahland.LandesKoord;

CLASS Pistenzustand =
  PraeparierteFlaeche: AhlandGebietseinteilung;
END Pistenzustand;
```

Da solche überlappungsfreien Flächen recht häufig vorkommen, wurde dafür in INTERLIS ein eigener Typ (AREA) eingeführt. Statt von Flächen wird von Gebietseinteilungen gesprochen.

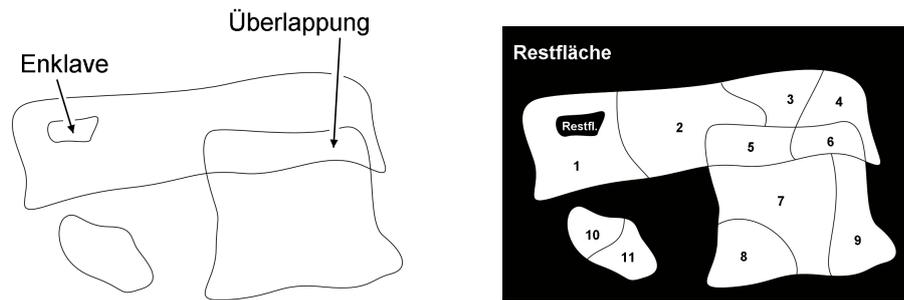


Abbildung 42: Beim gewöhnlichen Flächentyp (SURFACE, links) dürfen sich die Flächen verschiedener Objekte überlappen. Beispielsweise spricht nichts dagegen, wenn dasselbe Stück Land gleichzeitig zu zwei Skipisten gehört. Dagegen wird bei einer Gebietseinteilung (AREA, rechts) gefordert, dass jeder Punkt im Land eindeutig einem Objekt zugeordnet werden kann, wenn er nicht zur Restfläche (schwarz dargestellt) gehört. Ein Beispiel sind die Abschnitte, welche der Pistendienst präpariert.

6.9.5 Dreidimensionale Linientypen

Ist der zur Liniendefinition gehörige Koordinatentyp ein dreidimensionaler Typ, ist auch der Linientyp dreidimensional. INTERLIS 2 verzichtet dabei darauf, die dritte Dimension gleichberechtigt zu den ersten beiden zu führen, da in geographischen Anwendungen die drei Dimensionen immer in die Lage und eine Höheninformation aufgeteilt werden können.

- INTERLIS 2 unterstützt Linien mit 2.5 Dimensionen.

Dabei wird davon ausgegangen, dass jeder Stützpunkt (Punkt zwischen zwei Linienstücken) mit Lage und Höhe definiert ist und die Höhe auf dem Linienstück entsprechend der Länge des Kurvenstückteils linear interpoliert wird.

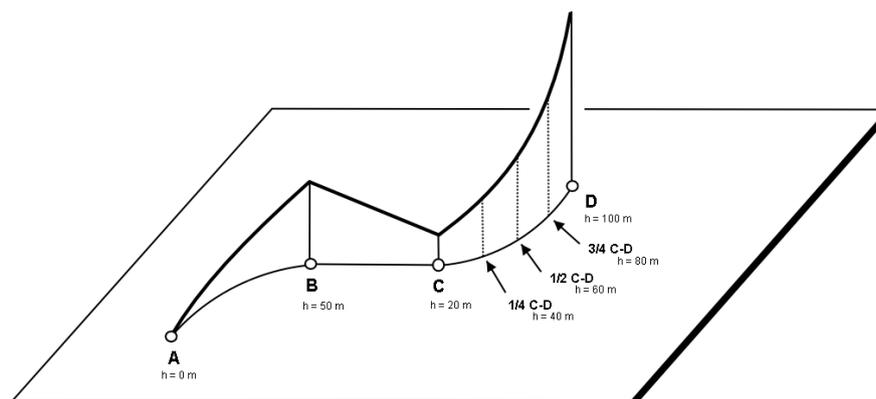


Abbildung 43: INTERLIS unterstützt 2.5-dimensionale Linien: Die Höhe zwischen zwei Stützpunkten wird immer linear interpoliert. An jener Stelle, wo auf dem Boden ein Viertel des Weges zwischen C und D zurückgelegt wurde, ist auch ein Viertel des Höhenunterschieds überwunden.

Sollte man den Pistenverlauf nun nicht mit einem dreidimensionalen Linientyp modellieren? Rein technisch wäre dies offenbar kein Problem, und die Höhe spielt schliesslich beim Skifahren eine wichtige Rolle. Dagegen spricht aber, dass die Höhe des Pistenverlaufs keine unabhängige Grösse ist: Kennt man die Lage, ergibt sich die Höhe aus der Geländeform. Die Höhe des Pistenverlaufs kann damit anhand seiner Lage und einem Geländemodell berechnet werden. Aus konzeptueller Sicht ist es darum vorzuziehen, auf die Höheninformation beim Pistenverlauf zu verzichten.

Anders kann es bei Strassen und Eisenbahnen sein, denn bei Brücken und Tunnels stimmt die Höhe nicht mit der Terrainhöhe überein. Allenfalls wird auch für die Höhe eine so grosse Genauigkeit gefordert, dass eine Ableitung aus dem Geländemodell nicht in Frage kommt. In gewissen Fällen kann es auch Sinn machen, die Kunstbauten (mit Höhe) unabhängig vom Trasseeverlauf zu modellieren. In diesem Fall würde die effektive Trasseehöhe im Bereich der Kunstbauten aus dem Modell errechnet; an den übrigen Stellen würde auf das Geländemodell zurückgegriffen.

Ein wichtiges Entscheidungskriterium in dieser Frage dürfte der Aufwand für Erfassung und Nachführung sein.

6.10 Wie bläst der Wind? – Strukturen

6.10.1 Mehrgliedrige Eigenschaften

Kurz vor der Ilisegg ziehen sich die Leute auf dem Sessellift vom Ilistäli die Kappe fest über die Ohren: Hier pfeift der Wind so richtig. Beim Wind ist eben nicht nur die Windgeschwindigkeit, sondern auch die Richtung aus der er kommt, massgebend. Führt man diese beiden Eigenschaften im Rahmen einer Klassenbeschreibung einfach zusammen mit anderen Attributen auf, kommt dieser Sachverhalt wenig zum Ausdruck.

```
CLASS Wetter =
  Temperatur: MANDATORY -50 .. 50 [oC];
  Windrichtung: MANDATORY (N, NE, E, SE, S, SW, W, NW) CIRCULAR;
  Windgeschwindigkeit: MANDATORY 0 .. 200 [kmh];
END Wetter;
```

Für Situationen, wo ein Sachverhalt nicht durch einen einzigen, sondern durch mehrere Werte beschrieben wird, ist es sinnvoll, eine Struktur (Windangabe) zu definieren, welche die verschiedenen Eigenschaften (Windrichtung, Windgeschwindigkeit) umfasst.

```
STRUCTURE Windangabe =
  Windrichtung: MANDATORY (N, NE, E, SE, S, SW, W, NW) CIRCULAR;
  Windgeschwindigkeit: MANDATORY 0 .. 200 [kmh];
END Windangabe;
```

Mit Struktur verwandte Begriffe sind: Datentyp, strukturierter Datentyp, ...

Überall, wo eine Aussage über den Wind gemacht wird, kann diese Struktur verwendet werden.

```
CLASS Wetter =  
  Temperatur: MANDATORY -50 .. 50 [oC];  
  Wind: Windangabe;  
END Wetter;  
  
CLASS Windmesser =  
  Standort: MANDATORY LandesKoord;  
  Wind: Windangabe;  
END Windmesser;
```

6.10.2 Mehrere Strukturelemente

Der Windmesser auf der Ilisegg ist noch etwas spezieller: Er zeigt nicht einfach den aktuellen Wert, sondern gleich die letzten sechs gemessenen Werte an. Deswegen werden die Ohren zwar nicht wärmer, aber erstaunlich ist es ja schon, wie schnell die Verhältnisse manchmal ändern können.

```
CLASS Windmesser =  
  Standort: MANDATORY LandesKoord;  
  Wind: LIST {6} OF Windangabe;  
END Windmesser;
```

Das Attribut Wind umfasst also sechs Elemente (Messwerte je mit Windrichtung und Windgeschwindigkeit). Mit LIST OF wird ausgesagt, dass die Reihenfolge relevant ist (z.B. der neuste Wert zuerst). Wäre die Reihenfolge nicht relevant, würde man BAG OF schreiben. Ähnlich wie bei den Beziehungen kann angegeben werden, wie viele Elemente minimal und maximal aufgeführt sein können.

6.10.3 Strukturen und Klassen

Strukturen und (Objekt-)Klassen sind einander formal sehr ähnlich. Sachlich bestehen aber erhebliche Unterschiede. Eine Klasse (Bahngesellschaft, Windmesser) beschreibt, wie Objekte aufgebaut sind. Eine Struktur beschreibt, wie kompliziertere Eigenschaften von Objekten (Windangabe) aufgebaut sind. Eine Struktur dient also dem gleichen Zweck wie ein Wertebereich, nämlich der Beschreibung wie ein Attribut aufgebaut ist. Manchmal bedarf es nur dann einer Struktur, wenn die Eigenschaft detaillierter beschrieben werden soll, während bei einfacher Beschreibung die Angabe eines Wertebereichs genügt (vgl. Abschnitt 6.12).

Die Instanzen von Klassen sind Objekte mit einer Eigenständigkeit (Ilisornbahnen, Windmesser auf der Ilisegg). Die Instanzen von Strukturen sind Strukturelemente (Wind mit 180 km/h aus NE). Der Wert eines Strukturattributes kann genau ein Strukturelement oder eine Menge von Strukturelementen (BAG OF, LIST OF) umfassen.

- Eine **Struktur** ist formal einer Objektklasse, sachlich einem Wertebereich sehr ähnlich. Die entsprechenden Exemplare, die **Strukturelemente**, haben aber keine eigene Identität, sondern sind Werte von Attributen eines Objektes.

Objekte können miteinander in Beziehung stehen (vgl. Abschnitt 6.13). Werte (von Wertebereichen oder Strukturen) können dies nicht. Man kann allerdings gleichartige Werte verschiedener Objekte (und verschiedener Klassen) miteinander vergleichen und so in eine Beziehung bringen (vgl. Abschnitt 6.17). So könnte man den Preis für den Wanderer-Pass

mit dem Preis für das Holzfällersteak vergleichen, das man im Falle des Fussmarsches im Bergrestaurant Ilishorn essen würde. Deswegen besteht aber keine Beziehung zwischen dem Wanderer-Pass und dem Holzfällersteak.

Für gewisse Fälle ist es nötig, dass zur Beschreibung einer Eigenschaft auf ein anderes Objekt verwiesen wird (vgl. Abschnitt 6.11.3). Auf einen Wert oder ein Strukturelement kann nie verwiesen werden, sie haben keine Identität.

6.10.4 Linien sind spezielle Strukturen

Das Attribut Verlauf der Piste (vgl. Abschnitt 6.9.1) ist als AhlandLinie, diese als POLYLINE definiert. Eine POLYLINE kann man als eine Menge von Linienstücken verstehen (vgl. Abschnitt 6.9.2). Die Definition als POLYLINE ist also nur eine abgekürzte Schreibweise für eine geordnete Menge von Strukturen, wobei die Strukturelemente einer bestimmten Strukturdefinition entsprechen:

```
STRUCTURE AhlandSegment (ABSTRACT) =
  SegmentEndPoint: MANDATORY Ahland.LandesKoord;
END AhlandSegment;

STRUCTURE AhlandStartSegment EXTENDS AhlandSegment (FINAL) =
END AhlandStartSegment;

STRUCTURE AhlandStraightSegment EXTENDS AhlandSegment (FINAL) =
END AhlandStraightSegment;

STRUCTURE AhlandArcSegment EXTENDS AhlandSegment (FINAL) =
  ArcPoint: MANDATORY Ahland.LandesKoord;
  Radius: Length;
END AhlandArcSegment;

CLASS Piste =
  Verlauf: LIST {2..*} OF AhlandSegment;
END Piste;
```

6.11 Wie wird im Ilistal gesprochen? – Mehrsprachigkeit

6.11.1 Pro Sprache ein Attribut

Im bisherigen Modell besitzt eine Bahngesellschaft einen Namen sowie eine Kurzbezeichnung. Wie kann damit erfasst werden, dass die Ilishornbahnen (IhB) auf Französisch *Remontées mécaniques de la Dent d'Illi (RDI)* heissen?

Es liegt nahe, die Objektklasse Bahngesellschaft um den französische Namen und die Kurzbezeichnung zu ergänzen:



Abbildung 44: Die Objektklasse Bahngesellschaft mit Namen und Kurzbezeichnung, jeweils auf Deutsch und Französisch.

Damit wäre der Fall klar. Was aber, wenn eines Tages der Wunsch aufkommen sollte, den Namen auch noch in einer dritten, vierten oder fünften Sprache zu erfassen? An sich kein Problem – es handelt sich ja nur um eine kleine Änderung des Datenmodells!



Es ist ja tatsächlich keine grosse Sache, ein Kästchen auf Papier um einige Zeilen zu erweitern. Wenn aber das Computersystem erst einmal realisiert worden ist, kann auch eine derart kleine Änderung aufwändig sein: Eingabeformulare wechseln, Programme müssen angepasst werden, Daten sind neu zu erfassen, usw.

6.11.2 Sprachabhängige Bezeichnungen als Strukturelemente

Besser ist es daher, wenn die konkrete Sprache gar nicht im Datenmodell vorkommt. In der folgenden neuen Version weist eine Bahngesellschaft eine Menge von Bahnbezeichnungen auf. Da der Umgang mit mehreren Sprachen ein häufiges Anliegen ist, erbt die Struktur Bahnbezeichnung die Grundstruktur Bezeichnung, welche die Sprache und einen Text umfasst.

```

STRUCTURE Bezeichnung =
  Name: TEXT;
  Sprache: TEXT*2;
END Bezeichnung;

STRUCTURE Bahnbezeichnung EXTENDS Bezeichnung =
  Name (EXTENDED): TEXT*100;
  Kurzbezeichnung: TEXT*10;
END Bahnbezeichnung;

CLASS Bahngesellschaft =
  Namen: BAG {1..*} OF Bahnbezeichnung;
END Bahngesellschaft;
  
```

Oder bildlich:

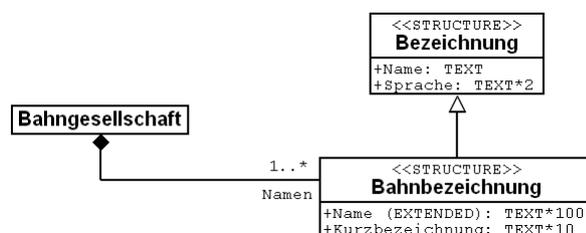


Abbildung 45: Einer Bahngesellschaft sind Bahnbezeichnungen zugeordnet. Da eine Gesellschaft mehrere Namen besitzen kann, ist es ohne Aufwand möglich, neue Namen in anderen

Sprachen aufzunehmen. Die Details der Zuordnung (Angaben wie 1..* oder das ausgefüllte Viereck) werden weiter unten im Zusammenhang mit Beziehungen diskutiert.



Zu beachten ist aber, dass nicht jedes Text-Attribut mehrsprachig sein muss. Familiennamen von Personen werden zum Beispiel nicht übersetzt.

Um Bezeichnungen in einer anderen Sprache hinzuzufügen, sind nur neue Daten zu erfassen. Das Datenmodell muss deswegen nicht angepasst werden.

6.11.3 Strukturelemente dürfen auf Objekte verweisen

Wer kennt schon die Sprachabkürzung für das Rätoromanische? rr? rm! Im Rahmen des nationalen Verbandes ist es klar, welche Sprachen für die Bezeichnungen der Bahngesellschaften in Frage kommen. Bei der Erfassung einer Bahn ist in der Regel nur eine Abkürzung zu beachten. Die kann man sich gut merken, weshalb der nationale Tourismusverband sein Modell wie oben beschrieben aufgebaut hat.

Wäre dies nicht der Fall gewesen, hätte man ein Modell gewählt, bei dem die Sprachen eigentliche Objekte sind. Das Sprachobjekt würde die Abkürzung und z.B. den Namen je als Text in der eigenen Sprache und in englisch enthalten.

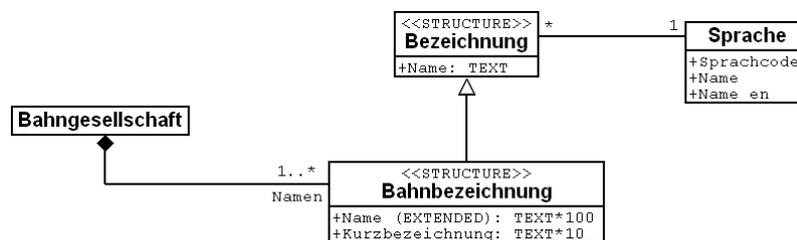


Abbildung 46: In dieser Variante verweist die (Sprach)Bezeichnung (eine Struktur) auf die Sprache (eine normale Objektklasse).

Die Bezeichnung verweist so auf die Sprache. Dieser Verweis ist aber nicht eine vollwertige Beziehung (vgl. Abschnitt 6.13), da die Bezeichnungen keine Identität haben. Aus Sicht des Sprachobjektes besteht darum auch kein direkter Zugang zu den Bezeichnungselementen. Dieser müsste über eine Sicht etabliert werden (vgl. Abschnitt 6.17).

6.12 Wie ticken die Ilistaler? – Modellierung von Zeit

6.12.1 Für einfache Ansprüche genügend

Der nationale Verband hat eine simple Lösung und für die Gültigkeitsdauer von Billettarten ein Attribut vorgesehen, das die Anzahl Tage (mit einer Stelle nach dem Komma) vorsieht.

```
Gultigkeitsdauer: 0.0 .. 1000.0 [d];
```

Wenn man es – wie die Ilistaler – etwas genauer nimmt, stellen sich verschiedene Fragen:

- ◆ Ein Billett, das am Ausgabetag gültig ist, hat doch nicht die gleiche Gültigkeit wie eines, das 24 Stunden gültig ist.
- ◆ Ein Monat hat mal 28, mal 30, mal 31 Tage.
- ◆ Ein Jahr mal 365, mal 366 Tage.

Auf Anfrage erhielten die Ilistaler vom nationalen Verband die Antwort, dass folgendes gelte:

- ◆ 0.9: am Ausgabetag;
- ◆ 30.0: einen Monat;
- ◆ 365.0: ein Jahr.



Solche Behelfslösungen sind manchmal verlockend, weil sie auf den ersten Blick einfach sind. Aber was, wenn mit 30.0 Tagen wirklich so viele Tage und nicht ein Monat gemeint ist? Darum ist Vorsicht angesagt!

Wie könnte aber eine bessere Lösung aussehen?

6.12.2 Zeitdauer als Struktur

Objekteigenschaften wie die Gültigkeitsdauer können nicht immer durch einen einzigen Wert genügend präzis beschrieben werden. Manchmal braucht es eine Gruppe von Attributen, manchmal macht es Sinn, verschiedene Erweiterungen vorzusehen. Dafür bietet sich die Struktur an.

```
STRUCTURE Zeitdauer (ABSTRACT) =
END Zeitdauer;

STRUCTURE ZeitdauerHeute EXTENDS Zeitdauer =
END ZeitdauerHeute;

STRUCTURE ZeitdauerInTagen EXTENDS Zeitdauer =
  Dauer: MANDATORY Tage [d];
END ZeitdauerInTagen;

....

CLASS Billettart =
  Gueltigkeitsdauer: Zeitdauer;
END Billettart;
```

Die Gültigkeitsdauer einer bestimmten Billettart ist mit einer Instanz (einem Strukturelement) der Struktur `ZeitdauerHeute`, `ZeitdauerInTagen`, `ZeitdauerInMonaten`, etc. beschrieben. Man könnte sogar noch etwas präziser modellieren und dafür sorgen, dass die Einheit einer expliziten Dauer (Tag, Monat, etc.) immer eine `Zeitdauer` sein muss und für implizite `Zeitdauern` (Woche, Saison, etc.) eine Aufzählung definieren:

```
STRUCTURE Zeitdauer (ABSTRACT) =
END Zeitdauer;

STRUCTURE ZeitdauerImplizit EXTENDS Zeitdauer =
  Dauer: MANDATORY (Tag, Woche, Monat, Jahr);
END ZeitdauerImplizit;

STRUCTURE ZeitdauerExplizit (ABSTRACT) EXTENDS Zeitdauer =
```

```

Dauer (ABSTRACT): MANDATORY NUMERIC [TIME];
END ZeitdauerExplizit;

STRUCTURE ZeitdauerInMinuten EXTENDS ZeitdauerExplizit =
  Dauer (EXTENDED): MANDATORY 0 .. 200 [Units.min];
END ZeitdauerInMinuten;

STRUCTURE ZeitdauerInTagen EXTENDS ZeitdauerExplizit =
  Dauer (EXTENDED): MANDATORY 0 .. 1000 [d];
END inTagen;

```

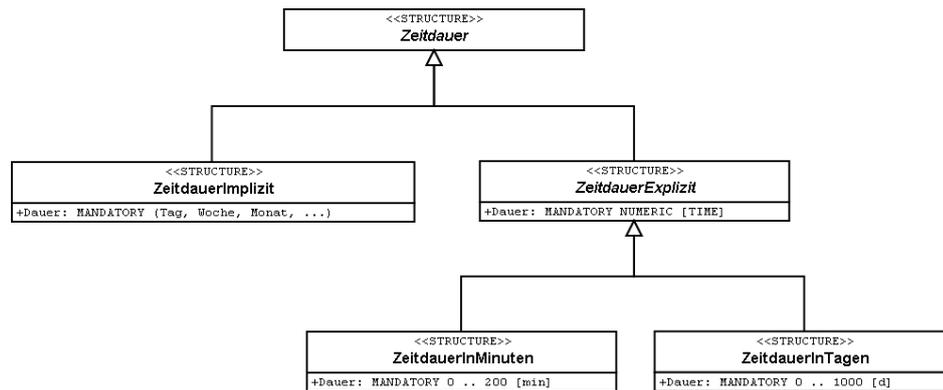


Abbildung 47: Zeitdauer in einer detaillierten Modellierung mit Strukturen. Damit ist es möglich, dass die Gültigkeitsdauer eines Billetts je nach Bedarf ein Monat (ZeitdauerImplizit; links) oder exakt dreissig Tage (ZeitdauerInTagen; rechts) ist.



Eine präzise, detaillierte, sachgerechte Modellierung ist grundsätzlich anzustreben. Man muss sich dabei aber immer bewusst sein, dass sie nur einen Sinn macht, wenn sie auch umgesetzt werden kann. Was bedeutet es für die Programmpakete? Und vor allem: Was bedeutet es für die Leute, die Daten erfassen und bearbeiten? Und umgekehrt: Was bedeutet es, wenn man vom möglichst korrekten Modell abweicht? Es kann also auch besser sein, sich mit der einfachen Lösung zufrieden zu geben.

6.12.3 Genaue Zeitdauer

Zeitdauern gibt es nicht nur für Billette. Die Ilistaler veranstalten jeden Freitag ein Skirennen für ihre Gäste. Dort werden Rennzeiten in Minuten, Sekunden und deren Bruchteilen gemessen. Es ist nahe liegend, dafür eine Struktur zu definieren, welche die Attribute Minuten und Sekunden aufweist:

```

STRUCTURE ZeitdauerInMinuten EXTENDS Zeitdauer =
  Minuten: 0 .. 9999.99 [min];
  Sekunden: 0.00 .. 59.99 [s];
END ZeitdauerInMinuten;

```

Damit der Zusammenhang zwischen Minuten und Sekunden ausgedrückt werden kann, bietet sich eine zusätzliche Möglichkeit an:

```

STRUCTURE ZeitdauerInMinuten EXTENDS Zeitdauer =
  Minuten: 0 .. 9999.99 [min];
  CONTINUOUS SUBDIVISION Sekunden: 0.00 .. 59.99 [s];
END ZeitdauerInMinuten;

```

Damit ist nichts darüber ausgesagt, in welcher Form solche Zeitdauern in einem Computer gespeichert werden. Es handelt sich nur um ein Mittel, möglichst konzeptnah zu beschreiben, was man wirklich will.

6.12.4 Formatierte Darstellung von Strukturen

Das Gästeskierennen wird immer so gestaltet, dass selbst die Skilehrer sicher mehr als dreissig Sekunden für die Abfahrt benötigen. Wer mehr als drei Minuten und dreissig Sekunden braucht, erhält am Ziel einen wärmenden Tee, die Zeit wird jedoch nicht festgehalten.

Wie kann nun der zulässige Wertebereich (30 Sekunden bis 3 Minuten und 30 Sekunden) festgehalten werden? Die Lösung liegt in formatierten Wertebereichen:

```

DOMAIN ZeitdauerinMinSec = FORMAT BASED ON ZeitdauerInMinuten
  ( Minuten ":" Sekunden );

CLASS Rennzeit =
  Vorname: TEXT*50;
  Name: TEXT*50;
  Laufzeit: FORMAT ZeitdauerinMinSec "0:30" .. "3:30";
END Rennzeit;

```

Ein formatierter Wertebereich nimmt auf eine Struktur Bezug und legt fest, wie aus den einzelnen Attributen der Struktur und aus Textkonstanten eine Zeichenkette entsteht, die den Wert wiedergibt. In dieser Form können Wertebereichseinschränkungen festgelegt werden. Die formatierte Darstellung wird auch für den Datentransfer verwendet. Damit ist es zum Teil möglich, gewisse extern verlangte Darstellungsformen direkt zu unterstützen. Dies kann insbesondere für die XML-konforme Darstellung von Zeitdauern und Zeitpunkten genutzt werden.

6.12.5 Zeitpunkte

Statusmeldungen über das Wetter, die Wartezeiten, die Pistenverhältnisse sollen im Ilistal immer mit dem Zeitpunkt versehen werden, in dem der Zustand festgestellt wurde. Erster Gedanke: Uhrzeit in Stunden und Minuten. Ja, damit man Statistiken erstellen kann, gehört natürlich noch das Datum dazu. Das sollte genügen!

Wirklich? Die Ilishornbahnen führen in schönen Vollmondnächten Sonderkurse zum Ilishorn, damit dort die beliebte Dracula-Party steigen kann. Da werden natürlich auch mitten in der Nacht Statusmeldungen geschickt. Auch um 2.30 Uhr. Auch an jenem Sonntag in der Früh, als von der Sommer- auf die Winterzeit umgestellt wurde. Allerdings gab es da ein grosses Durcheinander: Die neuste Meldung war plötzlich älter als die letzte! Natürlich: Alle Zeiten zwischen 2.00 und 3.00 gab es in jener Nacht doppelt, einmal gemäss Sommerzeit, einmal gemäss Winterzeit.

- ▶ Bei Zeitpunkten ist es immer wichtig zu wissen, welches das Bezugssystem ist.

Meinen wir Sommerzeit, Winterzeit, UTC? Je internationaler, desto wichtiger! Da kommt man schnell einmal auf den Gedanken, alles in UTC festzuhalten und es dem Computer zu überlassen, die Daten dem Benutzer gemäss seiner aktuellen Zeitzone zu präsentieren.

INTERLIS 2 bietet die Möglichkeit an, nicht nur den Wertebereich und die Einheit, sondern auch das Bezugssystem zu beschreiben. Für die UTC-Zeiten sind bereits formatierte Wertebereiche gemäss den XML-Regeln vordefiniert (XMLTime, XMLDate, XMLDateTime).

Gerade Öffnungs- oder Betriebszeiten werden aber vorzugsweise in der lokalen Zeit beschrieben. Mitternacht ist eben um 24.00 Uhr, unabhängig davon, ob gerade Sommer- oder Winterzeit ist. Dies sind aber nicht eigentliche Zeitpunkte. Vielmehr beschreiben sie Differenzen zu Mitternacht gemäss der aktuell gültigen Zeit.



Überall dort, wo die Zeit, vor allem aber wo Zeitpunkte von Bedeutung sind, ist höchste Aufmerksamkeit geboten.

6.13 Tarifbereiche, Zustandmeldungen – Beziehungen

6.13.1 Rollen

Was ist nun eine Bahngesellschaft für eine bestimmte Bergbahn? Eigentümerin? Betreiberin! In der Beziehung zwischen Bahngesellschaft und Bergbahn ist die Bahngesellschaft in der Rolle der Betreiberin.

Der Rollenname wird in der Grafik am Ende der Beziehungslinie auf der Seite des Inhabers der Rolle angeschrieben. Wenn er sich aber nicht vom Klassennamen unterscheidet, wird der Rollenname meistens weggelassen.

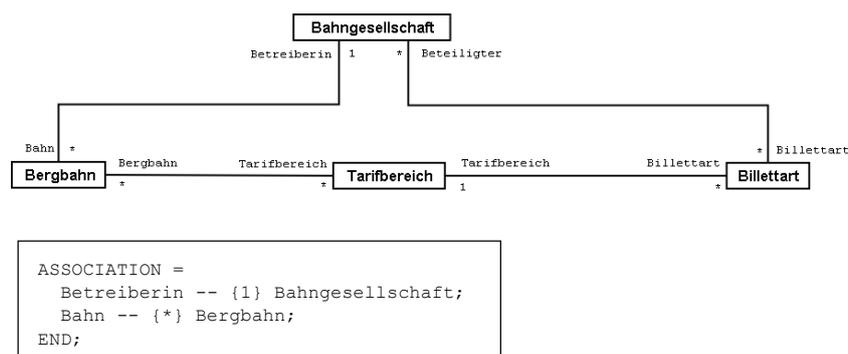


Abbildung 48: Gemäss diesem Modell ist es möglich, nach der Betreiberin einer Bergbahn zu fragen. «Betreiberin» ist eine *Rolle*, welche die Klasse «Bahngesellschaft» gegenüber der Klasse «Bergbahn» einnimmt. Unten ist die Beziehung zwischen Bahngesellschaft und Bergbahn in der Schreibweise von INTERLIS wiedergegeben.

Es ist durchaus normal, dass Rollennamen gewählt werden, die sich nicht von den Klassennamen unterscheiden. In der Beziehung Bergbahn – Tarifbereich macht es z.B. wenig Sinn, weitere Namen einzuführen. Allerdings ist der Bedarf nach zusätzlichen Namen ganz offen-

sichtlich gegeben, wenn eine Beziehung zwischen Objekten der gleichen Klasse besteht. So möchte man auch darstellen können, dass eine Bahngesellschaft andere Bahngesellschaften als Tochtergesellschaften besitzt.

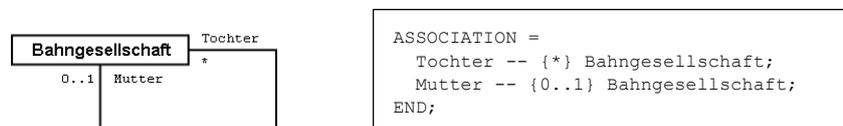


Abbildung 49: Eine Bahngesellschaft kann einerseits Mutter, andererseits auch Tochter einer anderen Bahngesellschaft sein. In solchen Fällen eignet sich der Klassenname nicht als Rollenname. Das Beispiel ist links in der graphischen Schreibweise von UML, rechts in der textuellen von INTERLIS wiedergegeben.

6.13.2 Stärke einer Beziehung

Assoziation, Aggregation und Komposition sind Ausdrücke für unterschiedliche Stärken von Beziehungen.

- ◆ **Assoziation** – Die Beziehung zwischen Tarifbereich und Bergbahn ist recht lose. Zwei Objekte sind einander zugeordnet, ohne dass eines dem anderen untergeordnet wäre. Die Assoziation ist eine Beziehung unter Gleichberechtigten. Meist sind in einem Datenmodell die Mehrzahl der Beziehungen gewöhnliche Assoziationen.
- ◆ **Aggregation** – Eine Bergbahn ist ein recht selbständiges Objekt. Es braucht aber immer eine Bahngesellschaft, um sie zu betreiben. Die Bahngesellschaft ist der Bergbahn übergeordnet.
- ◆ **Komposition** – Eine sehr enge Beziehung besteht zwischen einer Bergbahn und ihren Masten. Ein Mast macht eigentlich nur im Zusammenhang mit einer bestimmten Bergbahn einen Sinn. Die Komposition ist die Beziehung zwischen einem Ganzen und seinen (meist physischen) Bestandteilen.

Die Einteilung gemäss diesen Stärken ist nicht immer einfach. Aus Sicht der Informatik gibt es aber noch weitere Regeln, die den Entscheid manchmal vereinfachen:

- ◆ **Löschen** – Wird eine Bahngesellschaft gelöscht, hat das auf die zugeordneten Bergbahnen nur zur Folge, dass sie keine Betreiberin mehr haben. Wird jedoch eine Bergbahn gelöscht, werden auch alle Masten gelöscht. Das Löschen eines Ganzen entfernt auch alle Bestandteile, die mit ihm über eine Komposition verbunden sind.
- ◆ **Kopieren** – Wird eine Bahngesellschaft kopiert (in der Natur natürlich nicht so einfach wie im Computer), werden auch für alle zugeordneten Bergbahnen Kopien erstellt und der neuen Bahngesellschaft zugeordnet. Für jede Bergbahn werden entsprechend wieder Kopien der Masten erstellt. Das Kopieren eines Objekts erzeugt auch Duplikate jener Objekte, die ihm über Aggregationen und Kompositionen zugeordnet sind. Dagegen entstehen keine Kopien für jene Objekte, die ihm über gewöhnliche Assoziationen zugeordnet sind.

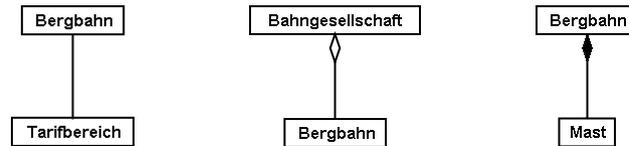


Abbildung 50: Assoziation (links), Aggregation (mitte) und Komposition (rechts) sind verschiedene Arten von Beziehungen. Sie unterscheiden sich in ihrer Bindungsstärke: Ein Mast ist so eng mit seiner Bergbahn verbunden, dass er als Bestandteil der Bahn aufgefasst werden kann. Im Vergleich zur Komposition sind Aggregation und Assoziation schwächer.

Die Schreibweise in INTERLIS ist der graphischen Darstellung nachempfunden. Der Rollenname muss jedoch auch dann geschrieben werden, wenn er sich nicht vom Namen der Klasse unterscheidet.

```
ASSOCIATION =
  Bergbahn -- Bergbahn;
  Tarifbereich -- Tarifbereich;
END;

ASSOCIATION =
  Betreiberin -<> Bahngesellschaft;
  Bergbahn -- Bergbahn;
END;

ASSOCIATION =
  Bergbahn -<#> Bergbahn;
  Mast -- Mast;
END;
```

6.13.3 Beziehungen mit Attributen

Diverse Billettarten berechtigen zur Fahrt auf Bergbahnen, die von verschiedenen Bahngesellschaften betrieben werden. Damit stellt sich die Frage, wie der mit dem Billettverkauf erzielte Erlös auf die einzelnen Gesellschaften verteilt wird. Zum Beispiel berechtigt das nationale Generalabonnement auch zur Fahrt mit der Ilishornbahn. Gemäss Abmachung erhalten die Ilishornbahnen dafür 0.13% des Umsatzes an Generalabonnements vergütet.

Beziehungen können auch Attribute aufweisen und haben so den Charakter von speziellen Klassen.

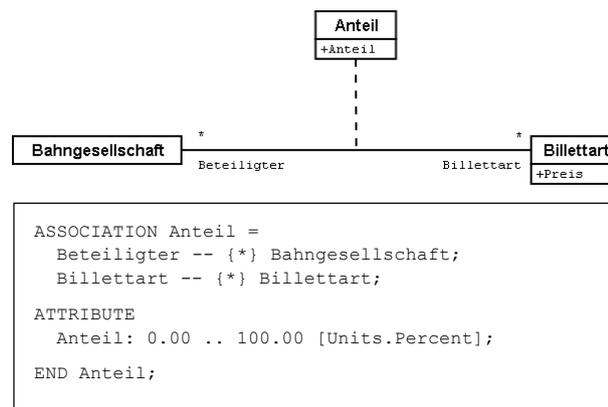


Abbildung 51: Eine Bahngesellschaft ist zu einem festgelegten Prozentsatz am Erlös aus dem Verkauf einer bestimmten Art von Billetten beteiligt. Der vereinbarte Anteil ist weder eine Eigenschaft der Bahngesellschaft noch der Billettart. Stattdessen handelt es sich um eine Eigenschaft ihrer Beziehung. Solche Situationen werden mit Beziehungsklassen modelliert.

6.13.4 Mehrgliedrige Beziehungen

Um einen besseren Überblick über die Billettverkäufe zu erhalten, möchte der nationale Verband in Zukunft noch festhalten, welche Verkaufsstelle von welcher Billettart in welcher Saison wie viele Exemplare verkauft hat.

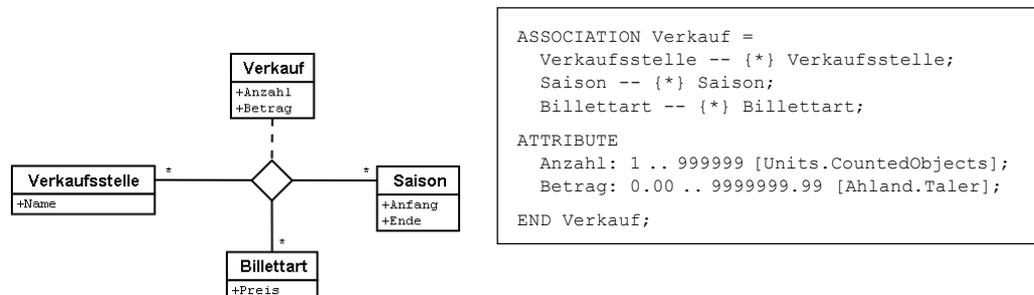


Abbildung 52: Der Verkauf wird pro Verkaufsstelle, Billettart und Saison erfasst. Es handelt sich um eine mehrgliedrige Beziehung zwischen drei gleichberechtigten Partnern (den Klassen Verkaufsstelle, Billettart und Saison). Dagegen ist «Verkauf» eine Beziehungsklasse, welche Eigenschaften der Beziehung (zum Beispiel die Anzahl verkaufter Billette sowie den umgesetzten Betrag) festhält.

Verkaufsstelle, Billettart und Saison stehen damit in einer gleichberechtigten Beziehung, auf der zudem noch die Anzahl der verkauften Billette und der umgesetzte Betrag als Attribut festgehalten sind. Diese Beziehung verbindet also nicht mehr zwei, sondern drei Klassen.

Was aber bedeuten bei mehrgliedrigen Beziehungen die Kardinalitätsangaben genau? Die Kardinalitätsangabe z.B. bei der Saison (*) sagt aus, dass es für eine bestimmte Kombination von Billettart und Verkaufsstelle beliebig viele Zuordnungen zu Saison-Objekten geben darf. Würde dort die Kardinalität 1 angegeben, würde das bedeuten, dass eine bestimmte

Billetart durch eine bestimmte Verkaufsstelle nur während einer Saison verkauft werden kann.

Etwas kompliziert. Braucht es wirklich mehrgliedrige Beziehungen, oder könnte man sie auf die üblichen Zweierbeziehungen reduzieren?

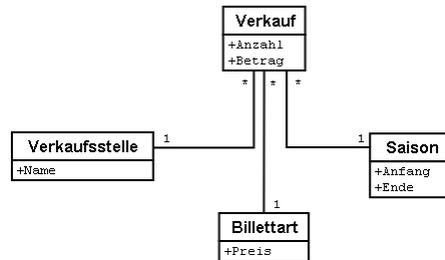


Abbildung 53: Beziehungen zwischen mehr als zwei Beteiligten lassen sich auf gewöhnliche Zweierbeziehungen reduzieren. Die frühere Beziehungsklasse (hier: Verkauf) wird zum gleichberechtigten Partner, und die Beteiligten stehen neu nur noch mit der früheren Beziehungsklasse in Beziehung.

Dieses Modell drückt aber weniger klar aus, dass die drei Klassen Verkaufsstelle, Billettart und Saison als Dreiergruppe miteinander in Beziehung stehen.

6.13.5 Geordnete Beziehungen

Betrachtet man alle Bergbahnen, die der Bahngesellschaft Ilishorn-Bahnen zugeordnet sind, ist damit keine bestimmte Ordnung verbunden. Die Frage, ob in der Zuordnung die Luftseilbahn vor oder nach der Gondelbahn erscheint, macht eigentlich gar keinen Sinn.

Natürlich kann man die Bahnen einer Gesellschaft in alphabetischer Reihenfolge auflisten. Diese Sortierung ist aber nicht eine Eigenschaft der Beziehung zwischen Bahngesellschaft und Bergbahn, sondern eine reine Frage der Darstellung. Für einen anderen Zweck könnte auch eine Sortierung nach Investitionskosten, Fahrzeit usw. interessant sein.

Aber wäre es nicht sinnvoll, mit der Ordnung festzuhalten, in welcher Reihenfolge die Beziehung etabliert wurde? Zuerst wurde die Luftseilbahn eröffnet, dann der Skilift, dann die Gondelbahn, usw. In diesem Fall wäre es allerdings besser, die Beziehung mit den Attributen Betriebsanfang und Betriebsende zu versehen. Dann könnte sogar festgehalten werden, welche Betreiberin es über die Zeit hinweg gegeben hat. Es macht in diesem Fall auch keinen Sinn mehr, die Beziehung als Aggregation aufzufassen.

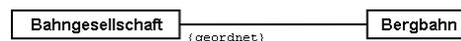


Abbildung 54: Um festzuhalten, in welcher Reihenfolge die Bergbahnen einer Gesellschaft ihren Betrieb aufgenommen haben, könnte man an sich eine geordnete Beziehung verwenden. Das Modell der nächsten Abbildung ist jedoch besser.

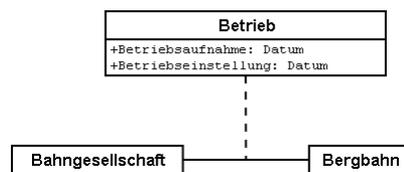


Abbildung 55: Das Modell ist mit einer Beziehungsklasse sauberer, weil damit weitere Auswertungen möglich sind. So können hier die Bahnen einer Gesellschaft auch anhand der Betriebs-einstellung sortiert werden, und ein Computerprogramm kann anzeigen, von wem eine Bergbahn in der Vergangenheit betrieben wurde.

Ähnliche Überlegungen gelten für die Beziehung zwischen Bergbahn und Masten: Mit der Ordnung der Beziehung könnte man die Reihenfolge von Tal- zu Bergstation ausdrücken. Konzeptuell ist es aber besser, beim Mast ein Lageattribut zu führen und die Reihenfolge dann aus dieser Lage und dem Trasseeverlauf abzuleiten.



Bevor eine Beziehung als geordnet deklariert wird, sollte genau überlegt werden, ob die Ordnung nicht aus Attributen der beteiligten Klassen oder der Beziehung abgeleitet werden kann.

Wo machen geordnete Beziehungen überhaupt einen Sinn? Die Gondelbahn von Ilisbad aufs Ilshorn hat Gondeln, die nicht fix auf dem Transportseil montiert sind. Sie können vielmehr in der Tal- und Bergstation abgestellt und je nach Bedarf ins Seil eingeklinkt werden. Welche Gondeln sind aktuell in welcher Reihenfolge auf das Seil eingeklinkt?

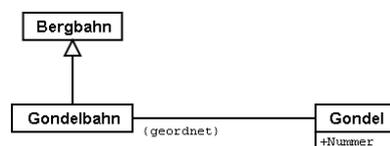


Abbildung 56: Zwar besitzt eine Gondel eine Nummer, aber diese sagt nichts über die Reihenfolge auf dem Seil aus. In diesem Fall ist eine geordnete Beziehung sinnvoll.

Hier ist die Ordnung gefragt. Die Nummer der Gondel kann nicht für die Ordnung herangezogen werden. Diese identifiziert einfach die konkrete Gondel. Über die Reihenfolge, wie sie aktuell auf dem Seil angeordnet sind, sagt sie überhaupt nichts aus.

6.13.6 Beziehungen erweitern

Eine Bahngesellschaft steht zu einer Reihe von Personen in Beziehung. Die einen sind bei ihr angestellt, die anderen an ihr beteiligt. Ähnlich wie zuvor bei den verschiedenen Arten von Bergbahnen gibt es auch hier verschiedene Möglichkeiten der Modellierung.

Eine Möglichkeit besteht darin, zwei unterschiedliche Beziehungen zwischen Bahngesellschaft und Person zu definieren: eine für die Anstellung, eine für die Beteiligung. Falls diese Unterscheidung einmal nicht wesentlich sein sollte (vielleicht für den weihnachtlichen

Versand eines Schoggi-Bähnlis), muss sich eine Anwendung aber um beide Beziehungen kümmern.



Abbildung 57: Eine Person kann gegenüber einer Bahngesellschaft Angestellter und/oder Beteiligter sein. Hier wird dies mit zwei verschiedenen Beziehungen modelliert. Will nun die Bahngesellschaft sowohl ihre Angestellten als auch ihre Teilhaber mit einem weihnachtlichen Schoggiversand beglücken, sind beide Beziehungen auszuwerten.

Eine andere Möglichkeit der Modellierung besteht darin, primär eine Beziehung zu definieren (Kontakt) und diese dann zu Anstellung bzw. Beteiligung zu erweitern. Solange es für eine Anwendung nicht relevant ist, in welcher Art des Kontaktes die Person zur Bahngesellschaft steht, nutzt sie die Kontakt-Beziehung und erhält so alle Personen, die in irgendeiner Art und Weise zur Bahngesellschaft in Kontakt stehen. Eine Anwendung, für die nur die Angestellten relevant sind, nutzt die erweiterte Beziehung Anstellung und erhält so nur die angestellten Personen.

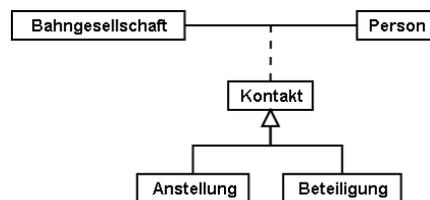


Abbildung 58: In dieser Variante ist die Beziehung zwischen Bahngesellschaft und Person allgemein mit der Beziehungsklasse «Kontakt» modelliert. Als Spezialfall eines Kontaktes gibt es auch die Anstellung und die Beteiligung. Wer nach den Kontakten der Gesellschaft fragt, wird automatisch auch die Angestellten und die Beteiligten erhalten. Beziehungsklassen sind also ähnlich wie Objektklassen erweiterbar, was im Diagramm wiederum mit einem weissen Pfeil gezeichnet wird.

Man könnte die Anstellungsbeziehung nochmals erweitern und z.B. eine Beziehung «Direktion» einführen.

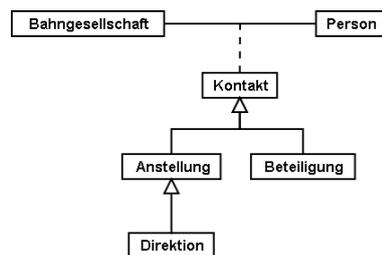


Abbildung 59: Die Beziehung zwischen einer Bahngesellschaft und ihrem Direktor («Direktion») ist ein Spezialfall der Beziehung «Anstellung».

Erweiterungen von Beziehungen gehen häufig Hand in Hand mit der Erweiterung von Objektklassen. Statt dass man z.B. von Anfang an sagt, eine Bergbahn weise Masten auf, spricht man zunächst nur von Betriebsmitteln. Diese sind der Bahn locker, also mittels Assoziation, zugeordnet. Da Masten eine wichtige Eigenschaft verschiedener Arten von Bergbahnen sind, wird die Klasse «BahnMitMasten» eingeführt. Diese hat eine Beziehung zu den Masten. Diese wird aber als Erweiterung der Beziehung zwischen Bergbahnen und Betriebsmitteln geführt. Da die Masten – etwa im Gegensatz zu einem Pistenfahrzeug – unmittelbar zu Bergbahn gehören, wird diese Beziehung zur Komposition. Die Stärke einer Beziehung darf in einer Erweiterung aber nur verstärkt, nicht aber gelockert werden, da sie sonst im Widerspruch zur Definition in der Basisdefinition stünde.

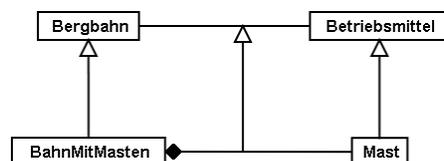


Abbildung 60: Bergbahn und Betriebsmittel führen eine allgemeine Beziehung, die von spezialisierten Klassen zur Komposition verstärkt wird.

6.13.7 Ableitbare Beziehungen

Wenn der Magen knurrt, wählt man lieber eine Skipiste, an der ein Gasthaus liegt. Deswegen müssen aber Pisten und Gasthäuser nicht miteinander in einer ständigen, ausdrücklichen Beziehung stehen. Es genügt zu wissen, dass das Gasthaus in der Nähe der Piste steht. Eine Aussage, die auf Grund von Lage des Gasthauses und Verlauf der Piste (je in Landeskoordinaten) herleitbar ist.



Nicht alles, was im Rahmen von Auswertungen zusammengehört, muss über Beziehungen verbunden sein. Gerade bei räumlichen Daten bilden die Koordinaten ein ideales Mittel, um Zusammenhänge bei Bedarf herzustellen.

Es macht auch keinen Sinn, sämtliche ableitbaren Beziehungen ins konzeptuelle Modell aufzunehmen. Darum fehlt die ableitbare Beziehung zwischen Gasthäusern und Pisten im konzeptuellen Modell.



Im konzeptuellen Modell sollen nur diejenigen impliziten Beziehungen beschrieben werden, die von konzeptueller Bedeutung sind. Darüber hinaus können die Programme natürlich weitere Beziehungen herstellen, indem sie die Attribute der Objekte geschickt (nicht zuletzt gemäss ihrer Lage) miteinander vergleichen.

Von konzeptueller Bedeutung sind nicht zuletzt Beziehungen, die in einigen Fällen explizit definiert werden müssen und in anderen Fällen ableitbar sind. Die Ableitung kann sich auf die Geographie oder auf andere Eigenschaften abstützen. Beispielsweise führten die Ilistaler einen speziellen Tarifbereich ein, der als Fläche umschrieben ist und alle Bergbahnen umfasst, deren Tal- und Bergstation innerhalb der Fläche liegt.

```
CLASS TarifbereichInGegend EXTENDS NatTour.Billette.Tarifbereich =
  Gegend: AhlandFlaeche;
END TarifbereichInGegend;
```

Die Beziehung zwischen diesem speziellen Tarifbereich und den Bergbahnen in der zugehörigen Gegend kann mit Sichten (vgl. Abschnitt 6.17) automatisch etabliert werden.

6.14 Einzigartige Ilishornbahnen – Konsistenzbedingungen

6.14.1 Grundsätzliches

Wir erinnern uns, dass die Ilishornbahnen für die einzelnen Bergbahnen jeweils den aktuellen Zustand erfassen wollen, unter anderem das Wetter bei der Bergstation:

```
CLASS Zustandsmeldung =
  Temperatur: MANDATORY -50 .. 50 [oC];
  Windrichtung: (N, NE, E, SE, S, SW, W, NW) CIRCULAR;
  Windgeschwindigkeit: MANDATORY 0 .. 200 [kmh];
  Wartezeit: ZeitdauerInMinuten;
  Erfasst: MANDATORY ZeitpunktMEZ;
END Zustandsmeldung;
```

Mit dieser Definition wäre auch eine Meldung mit einer undefinierten Windrichtung und einer Windgeschwindigkeit von 60 km/h erlaubt. Das ist aber nicht die Meinung. Eine undefinierte Windrichtung soll Windstille bedeuten. Dann aber ist die Windgeschwindigkeit natürlich 0. Und umgekehrt soll bei einer Windgeschwindigkeit grösser Null immer auch die Windrichtung definiert sein.

- ▶ Situationen, in denen zwischen verschiedenen Attributen eines Objektes oder gar zwischen verschiedenen Objekten ein bestimmter Zusammenhang bestehen muss, werden mit **Konsistenzbedingungen** beschrieben.

Eine Konsistenzbedingung wird in der Regel mit einer Formel beschrieben, deren Auswertung ergibt, ob die Bedingung erfüllt ist oder nicht. Mit einem solchen logischen Ausdruck kann die Windstille geregelt werden:

```

CLASS Zustandsmeldung =
  Temperatur: MANDATORY -50 .. 50 [oC];
  Windrichtung: (N, NE, E, SE, S, SW, W, NW) CIRCULAR;
  Windgeschwindigkeit: MANDATORY 0 .. 200 [kmh];
  Wartezeit: ZeitdauerInMinuten;
  Erfasst: MANDATORY ZeitpunktMEZ;
MANDATORY CONSTRAINT
  DEFINED (Windrichtung) == (Windgeschwindigkeit > 0);
END Zustandsmeldung;

```

Genau dann wenn die Windrichtung definiert ist, muss die Windgeschwindigkeit grösser Null sein. Ist die Windrichtung nicht definiert, muss die Geschwindigkeit Null sein. Es wird also gefordert, die «Definiertheit» der Windrichtung solle gleich (==) der «Positivheit» der Windgeschwindigkeit sein.

Häufig sind aber Konsistenzbedingungen vermeidbar, wenn das Modell anders aufgebaut wird. Packt man die Windangaben in eine Struktur und sagt, dass diese Struktur als Ganzes fakultativ ist, braucht es keine Konsistenzbedingung. Bei Windstille fehlt das Strukturelement. Windet es, sind zwingend Windrichtung und Windgeschwindigkeit vorhanden.

```

STRUCTURE Windangabe =
  Windrichtung: MANDATORY (N, NE, E, SE, S, SW, W, NW) CIRCULAR;
  Windgeschwindigkeit: MANDATORY 0 .. 200 [kmh];
END Wind;

CLASS Zustandsmeldung =
  Temperatur: MANDATORY -50 .. 50 [oC];
  Wind: Windangabe;
  Wartezeit: ZeitdauerInMinuten;
  Erfasst: MANDATORY ZeitpunktMEZ;
END Zustandsmeldung;

```



Bei Konsistenzbedingungen – insbesondere wenn sie kompliziert sind – besteht immer der Verdacht, dass das optimale Modell noch nicht gefunden wurde. Andererseits macht es auch keinen Sinn, ein an sich einfaches Modell künstlich kompliziert zu machen, nur um eine Konsistenzbedingung zu vermeiden.

6.14.2 Plausibilitätsbedingungen

Die Angestellten der Ilishornbahn verdienen allesamt nicht schlecht, aber der Direktor bezieht ein doch deutlich höheres Gehalt.

Konsistenzbedingungen gelten üblicherweise für alle Objekte der betreffenden Klasse. In INTERLIS 2 werden sie als MANDATORY CONSTRAINT bezeichnet. Man spricht gelegentlich von «harten» Bedingungen, weil sie immer erfüllt sein müssen. Nun gibt es aber auch Bedingungen, die in aller Regel, aber eben nicht immer erfüllt sind.

Bei Attributen wie Monatssalär oder Körpergrösse muss der grundsätzlich zulässige Bereich relativ gross gewählt werden. Die Werte der meisten Objekte liegen aber unterhalb einer wesentlich tieferen Grenze. Ausnahmen sind dennoch möglich, wie zum Beispiel für den Lohn des Bahndirektors.

```

ASSOCIATION Anstellung =
...
  Monatssalaer: MANDATORY 0 .. 20000 [Taler];
...
CONSTRAINT >= 95%
  Monatssalaer < 10000;
END Anstellung;

```

Es wird geschätzt, dass das Monatssalär in mindestens (\geq) 95% aller Fälle unter 10000 Talern liegt. Umfasst ein Modell solche «weichen» Bedingungen, wird es möglich, Daten bei der Eingabe auf ihre Plausibilität zu prüfen und statistisch zu kontrollieren.

6.14.3 Eindeutigkeitsbedingungen

Wie werden die Personen identifiziert, die bei Bergbahngesellschaften arbeiten oder an ihnen beteiligt sind? Nahe liegend, aber ungeeignet wäre es, hierzu Name und Vorname zu verwenden:

```

CLASS Person =
  Name: TEXT*50;
  Vorname: TEXT*20;
  UNIQUE Name, Vorname;
END Person;

```

Es wäre dann unzulässig, zwei verschiedene Personen mit der gleichen Kombination von Name und Vorname aufzuführen. Der neue Lokomotivführer Hans Huber darf seine Arbeit also erst antreten, wenn zuvor sein Namensvetter in der Buchhaltung entlassen wird.

Was wäre eine bessere Eindeutigkeitsbedingung? Warum überhaupt Eindeutigkeitsbedingungen?

- ▶ Eine **Eindeutigkeitsbedingung** dient nicht der Identifikation eines Objektes innerhalb eines Programmpakets. Stattdessen beschreibt es, welche Kombinationen von Attributen sachlich eindeutig sein müssen.
- ▶ Programm-intern und beim Datenaustausch ist ein Objekt durch eine technische **Objektidentifikation** gekennzeichnet. Diese hat keinerlei Bedeutung für die Anwendung.

Es braucht also nicht für jede Objektklasse eine Eindeutigkeitsbedingung, nur damit ein Objekt identifizierbar ist. Es reicht, wenn das Datenobjekt, welches der effektiven Person entspricht, bei der Eingabe gefunden werden kann. Hierfür können auch Attribute, Beziehungen usw. herangezogen werden, ohne dass eine Kombination eindeutig sein muss.

Will man jedoch eine system-externe Identifikation, die zum Beispiel für Menschen verständlich ist, braucht es ein Attribut oder eine Kombination von Attributen, deren Werte über alle Objekte eindeutig sind. Häufig werden dafür künstliche Attribute geschaffen (Versicherungsnummer, Kundennummer, Artikelnummer, usw.)



Künstliche Identifikatoren sind wo immer möglich zu vermeiden. Wo sie dennoch nötig sind, ist darauf zu achten, dass sie nicht Inhalte anderer Attribute in redundanter Form enthalten.

Bei den Ilishornbahnen wurde dieses Problem auf simple Weise gelöst: Man hat einfach eine Personalnummer eingeführt. Stösst eine Person neu zu den Ilishornbahnen, wird ihr eine Nummer zugewiesen, die noch nicht vergeben ist.

```
CLASS Person =  
  Name: TEXT*50;  
  Vorname: TEXT*20;  
  Personalnummer: 1 .. 9999;  
  UNIQUE Personalnummer;  
END Person;
```

Heikler würde die Sache, wenn die Klasse, durch die Personen beschrieben sind, nicht durch die Ilishornbahnen, sondern durch den nationalen Verband definiert wäre. Die Personalnummern aller Personen im Rahmen des Verbandes müssten dann eindeutig sein – auch wenn sie dezentral erfasst werden. Gäbe es zwei gleiche Nummern (z.B. eine bei den Ilishornbahnen, eine bei den Blaubergbahnen), wäre die Bedingung verletzt.

- Eindeutigkeitsbedingungen gelten immer für alle Objekte, die der Klasse entsprechen, für welche die Bedingung gilt – auch wenn die Entsprechung nur indirekt (in Form einer Erweiterung der Klasse) ist.

Eine Bahngesellschaft kann mehrere Namen tragen. Es soll aber pro Sprache nur eine einzige Bahnbezeichnung geben können; die Ilishornbahnen dürfen somit keinen zweiten deutschsprachigen Namen führen. Allerdings gilt diese Einschränkung nur lokal, also pro Gesellschaft: Schliesslich besitzen ja auch die Blaubergbahnen einen deutschsprachigen Namen. Über alle Gesellschaften hinweg gesehen gibt es also durchaus mehr als einen Namen in derselben Sprache. Die Sprache von Bahnbezeichnungen muss nur für eine bestimmte Bahngesellschaft eindeutig sein.

- Weist ein Objekt Unterstrukturen auf, soll die Eindeutigkeit – anders als bei den eigentlichen Objekten – in der Regel nicht «global» für die Elemente von sämtlichen Unterstrukturen gelten. Sie bezieht sich meist nur «lokal» auf die Unterstrukturelemente von einem einzigen Objekt.

```

STRUCTURE Bezeichnung =
  Name: TEXT*100;
  Sprache: TEXT*2;
END Bezeichnung;

STRUCTURE Bahnbezeichnung EXTENDS Bezeichnung =
  Kurzbezeichnung: TEXT*10;
END Bahnbezeichnung;

CLASS Bahngesellschaft =
  Namen: BAG {1..*} OF Bahnbezeichnung;
UNIQUE
  (LOCAL) Namen : Sprache;
END Bahngesellschaft;

```

Wie aber erreicht man, dass die Kurzbezeichnungen der verschiedenen Bahnen nicht kollidieren? Sowohl die Blaubergbahnen wie die Buntbergbahnen möchten doch primär einmal BBB heissen. In INTERLIS 2 können Konsistenzbedingungen nicht nur für Objektklassen bzw. lokale Strukturelemente, sondern auch für Sichten (vgl. Abschnitt 6.17) formuliert werden. Mit einer bestimmten Sicht können aus Strukturelementen quasi eigenständige Objekte gemacht werden. Für diese kann dann wieder eine Eindeutigkeitsbedingung formuliert werden.

6.14.4 Existenzfordernis

Im Gegensatz zu Zahnrad- und Standseilbahnen ist der Trasseeverlauf von Luftseilbahnen, Gondelbahnen, Skiliften usw. nicht einfach beliebig, sondern an die Tal- und Bergstation sowie an die Masten gebunden.

Diesen Zusammenhang möchte man ausdrücken. Die Linien von INTERLIS 2 verbinden jedoch Stützpunkte, die primär Koordinaten sind und keinen Bezug zu Modellobjekten wie etwa Masten aufweisen. Der Zusammenhang zwischen dem Trasseeverlauf und anderen Objekten kann aber als Konsistenzbedingung formuliert werden.

Mit der folgenden Definition muss sich jeder Punkt des Trasseeverlaufs auf die Lage eines Masts (Mast:Lage), die Lage der Talstation einer Bergbahn (Bergbahn:LageTalstation) oder (OR) die Lage der Bergstation einer Bergbahn (Bergbahn:LageBergstation) abstützen.

```

CLASS BodenunabhaengigeBahn EXTENDS Bergbahn =
EXISTENCE CONSTRAINT
  Trasseeverlauf REQUIRED IN
    Mast:Lage
    OR
    Bergbahn:LageTalstation
    OR
    Bergbahn:LageBergstation;
END BodenunabhaengigeBahn;

```

Solche Existenzbedingungen sind nicht nur im Zusammenhang mit Linien, sondern auch bei gewöhnlichen Attributen formulierbar. Konzeptuell können sie immer als eine schwache Form einer Beziehung betrachtet werden.

6.14.5 Vererbung von Konsistenzbedingungen

Bereits bei der Bergbahn selbst wurde eine Konsistenzbedingung formuliert: Der Trasseeverlauf muss bei der Talstation beginnen und bei der Bergstation enden. Oder anders gesagt, der erste Punkt des Trasseeverlaufs (Trassee -> Segments[FIRST] -> SegmentEndPoint) muss gleich der Lage der Talstation sein, und (AND) der letzte Punkt des Trasseeverlaufs (Trassee -> Segments[LAST] -> SegmentEndPoint) muss mit der Lage der Bergstation zusammenfallen.

Abschnitt 7.3 erläutert, wie Linien aufgebaut sind. Dort wird auch das Attribut SegmentEndPoint besprochen, das für den Endpunkt eines Liniensegments steht.

```

CLASS Bergbahn =
  LageTalstation: Ahland.LandesKoord3;
  LageBergstation: Ahland.LandesKoord3;
  Trasseeverlauf: Ahland.LinieNormal;
MANDATORY CONSTRAINT
  Trassee -> Segments[FIRST] -> SegmentEndPoint == PARENT == LageTalstation
  AND
  Trassee -> Segments[LAST] -> SegmentEndPoint == PARENT == LageBergstation;
END Bergbahn;

```

Was bedeutet eine solche Definition für allfällige Erweiterungen dieser Klasse?

- ▶ Konsistenzbedingungen können durch Klassenerweiterungen nicht ausser Kraft gesetzt werden. Erweiterungen können nur zusätzliche Bedingungen definieren.

6.15 Wie eng gehören Betriebsentscheide zur Bahn? – Unabhängige Themen

6.15.1 Allgemeines

Betriebsentscheide beziehen sich jeweils auf eine bestimmte Bergbahn. Entsprechend sind die beiden Klassen durch eine Assoziation verbunden.

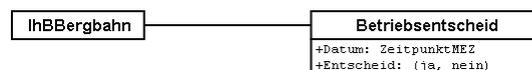


Abbildung 61: Die Klassen IhBBergbahn und Betriebsentscheid sind durch eine Assoziation verbunden.

Dennoch sind die Objekte der beiden Klassen recht unterschiedlich. Es braucht einiges, bis eine Bergbahn erstellt ist und bis sich irgendwelche Eigenschaften ändern. Solche Änderungen (und auch die der Billette) werden immer durch die Direktion beschlossen. Betriebsentscheide werden jedoch täglich durch den Betriebsleiter gefällt.

Noch extremer ist es mit den Zustandsmeldungen: Bei den wichtigeren Bergbahnen werden sie sogar automatisch alle zwanzig Minuten erzeugt. Zum Erfassen und Bearbeiten der Daten werden teilweise auch unterschiedliche Programmpakete eingesetzt. Diesen Sachverhalt möchte man im Konzept irgendwie darstellen.

- ▶ **Themen** (Topics) ordnen die Modelldefinition gemäss **Zuständigkeiten und zeitlichem Verhalten**.

Damit eröffnet sich die Möglichkeit, dass auf einem bestimmten Computersystem nicht alle Daten vorhanden sein müssen, oder dass bestimmte Themen nur gelesen, nie aber verändert werden.

- ▶ Zu einem Thema können mehrere **Behälter** existieren, die Daten enthalten, die diesem Thema entsprechen.

Das Computersystem der Ilishornbahnen umfasst z.B. je einen Behälter für die Bergbahnen, die Billette und die verschiedenen betrieblichen Aspekte. Der nationale Verband führt auch je einen Behälter für die Bergbahnen und die Billette. Die Ilishornbahnen schicken dem Verband immer die Änderungen, die sich in den Behältern der Bergbahnen und der Billette ergeben haben. Die Blauhornbahnen und alle weiteren Bahngesellschaften schicken dem Verband ebenfalls die Änderungen oder periodisch eine Kopie ihrer Datenbehälter. Der Verband integriert die Daten dieser Behälter dann in die eigenen Behälter.

Dank der Gliederung der Modelle in verschiedene Themen können die Daten gezielt geliefert werden. Es brauchen nur die Behälter jener Themen übermittelt werden, die für den Empfänger relevant sind.

6.15.2 Unabhängigkeit von Themen

Wird eine Bergbahn abgebrochen, wird als Folge das Datenobjekt gelöscht. Die Änderung wird nun auch dem nationalen Verband mitgeteilt. Wenn dabei aber nur der Behälter der Bergbahnen übermittelt wird, entsteht beim nationalen Verband ein Widerspruch in den Daten: Es gibt noch Tarifbereiche, die mit der Bergbahn verbunden sind, obwohl diese gelöscht wurde. Offensichtlich sind Beziehungen, die über Themengrenzen hinweg gehen, besonders heikel.

- ▶ Themen sollen möglichst wenig voneinander abhängen. Beziehungen, die über Themengrenzen hinwegführen, sind möglichst zu vermeiden. Im Modell sind sie speziell zu kennzeichnen.

In grafischen Darstellungen von Modellen sind solche Beziehungen relativ einfach erkennbar, wenn die Darstellung die Themen und die Beziehungen klar zeigt. In der textlichen Darstellung von INTERLIS 2 muss der Sachverhalt mit dem Schlüsselwort EXTERNAL gekennzeichnet werden. Sie sind zudem nur zulässig, wenn die Themen als voneinander abhängig deklariert wurden (DEPENDS ON). Gegenseitige Abhängigkeiten (auch indirekte) sind unzulässig.

Aber wie sollen Beziehungen über Themengrenzen vermieden werden, ohne als Folge einfach alles im selben Thema zu halten?

6.15.3 Die Verantwortung von Sender und Empfänger

Die Beziehung zwischen Betriebsentscheid und Bergbahn, auf den er sich bezieht, lässt sich nun aber mal nicht vermeiden. Dennoch macht es Sinn, die Bergbahnen und die Betriebsentscheide in verschiedenen Themen zu halten. Und bei dieser Beziehung ist es

auch kein Problem, dass die Dinge nicht zusammenpassen. Beide Themen und ihre zugehörigen Behälter werden ja durch die Ilishornbahnen nachgeführt. Vor allem bei schnelllebigen Objekten, die über Beziehungen in verschiedenen Themen angesprochen werden, sind Konflikte aber nicht immer auszuschliessen.

INTERLIS 2 legt dabei folgende Regelung fest:

- ▶ Die Korrektheit der Beziehungen innerhalb eines Behälters liegt in der Verantwortung des Senders. Der Empfänger muss damit fertig werden, dass Objekte, die an einer themenübergreifenden Beziehung beteiligt sind, zu einem bestimmten Zeitpunkt nicht bekannt sind. Der Empfänger darf aber davon ausgehen, dass auch die themenübergreifenden Beziehungen stimmen, wenn die Versionen der beteiligten Behälter zusammenpassen.

Die erste Regel, wonach ein Behälter in sich korrekt sein muss, ist auch dann zu beachten, wenn ein Behälter aus irgendeinem Grund aufgeteilt wird.

6.16 Alles Gute kommt von oben – Vorhandene Modelle nutzen

Die Ilistaler haben es gut gemacht: Sie haben nicht alles neu erfunden, sondern vorhandene Modelle (Units, Ahland, Adressen, NatTour) genutzt. Aber ein bisschen gefrevelt haben sie doch. Beispielsweise ist die AhlandLinie sicher nicht typisch für das Ilistal und seine Bergbahnen, sondern würde eigentlich in das landesweite Datenmodell von Ahland gehören. Auch WGS84Koord und die Tageszeit sind keine Ilistaler Spezialität.

Die Ilistaler haben sich insofern normal verhalten, als sie Dinge, die sie nicht anderswo gefunden haben, dem eigenen Modell beigefügt haben. Verständlich – aber nicht optimal.



Fehlende oder falsche Definitionen auf allgemeinerer (höherer) Stufe sollte man nicht einfach hinnehmen, sondern in Zusammenarbeit mit den zuständigen Stellen versuchen, diese Modelle zu verbessern.

Es ist darum auch sinnvoll, dass grundlegende Typen auf verschiedenen Stufen vordefiniert werden. INTERLIS selbst stellt einige Basismodelle zur Verfügung. Andere Modelle werden durch Anwendergemeinschaften (z.B. Verbände) bereitgestellt. Nochmals andere sind sehr spezifisch, wie etwa dasjenige der Ilishornbahnen.

Abschnitt 3.3 nennt einige Bezugsquellen für standardisierte Datenmodelle.

6.17 Tarifbereiche interessieren nicht – Sichten

6.17.1 Allgemeines

Wenn im Rahmen der Modellierung von Sichten gesprochen wird, ist natürlich nicht die Aussicht vom Ilishorn mit dem schönen Blick auf das Krummhorn gemeint. Und doch gibt es eine Ähnlichkeit zwischen den beiden Arten von Sichten. Auf der topographischen Karte sind das Ilishorn, das Krummhorn und all die weiteren Berge, Täler und Dörfer eingezeichnet und dabei die Höhe mit Zahlen und Höhenschichtlinien verdeutlicht. Die Karte zeigt die Aussicht vom Ilishorn nicht direkt. Sie enthält aber die nötigen Angaben, aus denen geübte

Kartenleser die Aussicht vom Ilishorn ableiten. Auf Grund der Karte ist klar, dass der Spitz, der links hinter dem Krummhorn hervor schaut, der «Schwarze Zahn» ist.

In der Analogie entsprechen die Objektklassen, Strukturen und Beziehungen eines Modells der Karte. Sie sind ein sachgerechtes Abbild der Realität, ohne dass ein bestimmter Verwendungszweck vorgegeben ist. Die Sichten eines Modells entsprechen der Aussicht vom Ilishorn. Sie unterstützen einen bestimmten Verwendungszweck. Sie nehmen dazu auf die Grundlagen oder andere Sichten Bezug und formen sie so um, dass der Verwendungszweck möglichst gut unterstützt werden kann.

Aber warum sollen dann solche Sichten Bestandteil des Modells sein? Im Modell soll doch nicht vorweggenommen werden, ob die Aussicht quasi vom Ilishorn, dem Krummhorn, vom Schwarzen Zahn oder vom Sprudelbecken im Garten des Kurhauses Iisbad aus genossen wird.

Vor allem für spezielle Konsistenzbedingungen (vgl. Abschnitt 6.14.3) und für ableitbare Beziehungen (vgl. Abschnitt 6.13.7) machen Sichten auch im Bereich des Modells Sinn. Sichten sind aber auch nützlich, wenn die Daten für einen bestimmten Zweck in einer aufbereiteten Form geliefert werden müssen, wie das bei der Datenabgabe zuhanden des Ilistaler Webdienstes der Fall ist. INTERLIS 2 bietet zudem noch die Möglichkeit, Grafiken zu definieren. Die Grundlage für die Grafikdefinition bilden in vielen Fällen nicht die Originaldaten, sondern Sichten.

- ▶ **Sichten** bauen auf Objektklassen oder anderen Sichten auf und kombinieren die Ausgangsobjekte auf verschiedene Arten zu neuen Sichtobjekten.

Die Sichten von INTERLIS sind mit den Sichten von Datenbank-Systemen vergleichbar.

6.17.2 Das Bildungsgesetz von Sichten

Der detaillierte Trasseeverlauf interessiert vielleicht nicht, wohl aber die Länge. Bei Personen ist man manchmal eher am Alter als am Geburtsjahr interessiert. Diese Eigenschaften können aus anderen abgeleitet werden. Würden derartige «redundante» Eigenschaften als normale Daten erfasst, wäre die Gefahr sehr gross, dass sie nicht immer auf dem aktuellen Stand wären. Das Alter einer Person ändert sich schliesslich jedes Jahr!

- ▶ Die wesentlichste Eigenschaft einer Sicht ist ihr **Bildungsgesetz**. Es legt fest, wie aus den Ausgangsobjekten abgeleitete Sichtobjekte entstehen.

Beispielsweise ist die Sicht «PersonMitAltersangabe» von der Objektklasse «Person» abgeleitet, der so genannten Basis. Eine einzelne PersonMitAltersangabe trägt dieselben Eigenschaften (ALL OF) mit genau denselben Werten wie die ursprüngliche Person. Zusätzlich fügt die Sicht aber noch eine weitere Eigenschaft «Alter» hinzu. Das Alter bestimmt sich (:=) aus der Differenz von Jahrgang und laufendem Jahr.

```

VIEW PersonMitAltersangabe
  PROJECTION OF Person;
=
  ALL OF Person;
  Alter: 0 .. 150 [y] := Difference (Person -> Jahrgang,
    PARAMETER laufendesJahr);
END PersonMitAltersangabe;

```

In diesem Beispiel gibt es für jedes Objekt der Basisklasse genau ein virtuelles Sicht-Objekt, also zu jeder Person eine entsprechende PersonMitAltersangabe.

- Das einfachste Bildungsgesetz einer Sicht ist die **Projektion (PROJECTION)**. Sie baut auf der Basis auf, übernimmt einzelne (oder auch alle) Attribute in einer beliebigen Reihenfolge und kann weitere, abgeleitete Attribute beifügen. Sie dient damit vor allem dazu, Attribute der bisherigen Objekte in eine nutzungsfreundliche Form zu bringen.

Der nationale Tourismusverband hat eine abstrakte Klasse «Tarifbereich» definiert. Im Ilistal will man jedoch nicht einzeln aufzählen, welche Bahnen an welchem Tarifbereich teilnehmen. Stattdessen gibt es dort räumlich umgrenzte Tarifbereiche, die im Modell mit der Klasse «TarifbereichInGegend» beschrieben sind. Diese Klasse besitzt eine Eigenschaft «Gegend» für das geographisch umgrenzte Gültigkeitsgebiet.

Eine Bahn, deren Tal- und Bergstation innerhalb der Gegend eines solchen räumlichen Tarifbereichs liegt, akzeptiert automatisch dessen Billette.

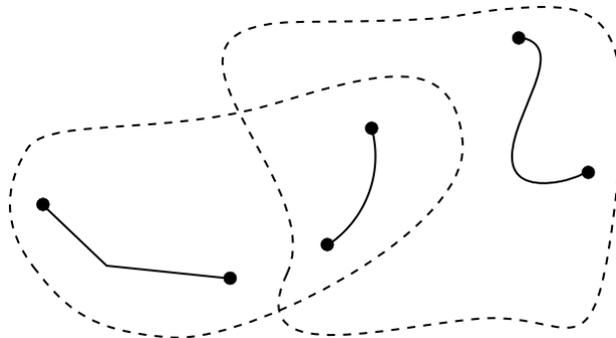


Abbildung 62: Welche Bergbahn liegt in der Gegend von welchem Tarifbereich? Die Ilistaler interessieren sich für alle Paare aus Bergbahn Bb und TarifbereichInGegend T, bei denen zwei Bedingungen gelten: Die Talstation von Bb muss in der Gegend von T liegen, und ebenso muss auch die Bergstation von Bb in der Gegend von T liegen.

Aber welche Bergbahn liegt nun konkret in der Gegend von welchem Tarifbereich? Auch diese Verbindung von Bergbahn und Tarifbereich kann mit einer Sicht abgeleitet werden.

- Das wohl wichtigste Bildungsgesetz einer Sicht ist die **Verbindung (JOIN)**. Sie verbindet mehrere Basisobjekte zu einem Sichtobjekt. Die Verbindung ist insbesondere als Basis für abgeleitete Beziehungen von Bedeutung.

```

VIEW BergbahnenInGegend
  JOIN OF Bb ~ Bergbahn,
        T ~ TarifbereichInGegend;
  WHERE InSurface(Bb -> LageTalstation, T -> Gegend) AND
        InSurface(Bb -> LageBergstation, T -> Gegend);
=
END BergbahnenInGegend;

```

Mit der Verbindung werden zunächst sämtliche Paare gebildet: Jedes Objekt der Klasse Bergbahn wird mit jedem Objekt der Klasse TarifbereichInGegend zu einem virtuellen Sichtobjekt verbunden.

Mit dem WHERE-Teil wird die Menge aller Sichtobjekte auf jene eingeschränkt, bei denen die Bedingungen erfüllt sind. Damit bleiben nur jene Paare aus Bergbahn *Bb* und Tarifbereich *T* übrig, wo Tal- und Bergstation von *Bb* im Gebiet von *T* liegt. In der obigen Abbildung erfüllen von den sechs möglichen Paaren (drei Bahnen x zwei Tarifbereiche) vier die Bedingung.

Bergbahn <i>Bb</i>	Tarifbereich <i>T</i>	Tal- und Bergstation von <i>Bb</i> in der Gegend von <i>T</i> ?
		ja
		ja
		nein
		nein
		ja
		ja

Abbildung 63: Betrachtet man in der letzten Abbildung sämtliche Kombinationen aus Bergbahn *Bb* und Tarifbereich *T*, so liegt nur bei vier der sechs Paare die Berg- und Talstation von *Bb* in der Gegend von *T*.

In einem letzten Schritt wird wie bei einer Projektion bestimmt, welche Eigenschaften die Sichtobjekte tragen sollen, und wie sich ihre Werte bestimmen. In der obigen INTERLIS-Definition dient hierzu der Teil nach dem Gleichheitszeichen.

Gibt es zu einer Bergbahn keinen Tarifbereich, kommt sie in der Sicht nicht vor. Mit einer besonderen Verbindung (so genannte «**Outer Join**») wird verlangt, dass auch dann ein Sichtobjekt entstehen soll, wenn zu einer Bergbahn kein Tarifbereich besteht. Für die konkrete Anwendung der Bergbahnen und Tarifbereiche macht dies allerdings kaum einen Sinn.

Möchte man ein Verzeichnis aller Koordinaten von Tal- und Bergstationen, steht dem die Tatsache entgegen, dass diese Koordinaten als einzelne Attribute der Bergbahnen aufbewahrt sind. Mit der **Vereinigung (UNION)** können sie aber zu einer Menge von gleichberechtigten Sichtobjekten zusammengefasst werden.

```
VIEW StationsKoordinaten
  UNION OF Talstation ~ Bergbahn, Bergstation ~ Bergbahn;
=
  Koordinaten: Ahland.LandesKoord := Talstation -> LageTalstation,
                                     Bergstation -> LageBergstation;
END StationsKoordinaten;
```

Die Menge der Sichtobjekte ist hier gleich der doppelten Menge der Bergbahnen. Einmal werden sie unter dem Aspekt Talstation, einmal unter dem Aspekt Bergstation ausgewertet. Das Attribut wird je nachdem gemäss dem Lageattribut von Tal- bzw. Bergstation gesetzt.

Zusammenfassung (AGGREGATION) und Aufschlüsselung (INSPECTION) haben mit Strukturattributen zu tun. Eine Zusammenfassung fasst Objekte, die bestimmte gleiche Eigenschaften aufweisen, zu einem einzigen Objekt zusammen. Im Rahmen des Sichtobjekts stehen die bisherigen Objekte als Elemente eines Strukturattributes zur Verfügung (vgl. Abschnitt 6.17.3). Eine Aufschlüsselung sorgt umgekehrt dafür, dass aus Strukturelementen eigenständige Sichtobjekte werden (vgl. Abschnitt 6.14.3).

6.17.3 Schrittweiser Aufbau von Sichten

Für die Billettkontrolle muss man bei jeder Bergbahn wissen, welche Billettarten gültig sind. Man möchte doch ein Verzeichnis aller Bergbahnen, in dem bei jeder Bergbahn die gültigen Billettarten aufgeführt sind. Losgelöst von den Basisdaten möchte man etwa folgendes Modell definieren:

```
CLASS Billettart =
  Namen: BAG {1..*} OF Bezeichnung;
  Preis: 0.00 .. 5000.00 [Ahland.Taler];
  Gueltigkeitsdauer: Zeitdauer;
END Billettart;

CLASS Bergbahn =
  Namen: BAG {1..*} OF Bezeichnung;
  GueltigeBillettarten: BAG OF Billettart;
END Bergbahn;
```

Aber wie kann das nun aus den Originaldaten abgeleitet werden? Das ist gar nicht so einfach. Einer Bergbahn können mehrere Tarifbereiche zugeordnet sein, welchen ihrerseits wieder mehrere Billettarten zugeordnet sind. Zudem gibt es Tarifbereiche, die alle Bergbahnen einer Gegend umfassen.

Der letzte Aspekt ist zum Glück schon erledigt, weil es eine abstrakte Beziehung zwischen Bergbahn und Tarifbereich gibt, die «Gültigkeit». Sie wird einerseits durch eine explizite Beziehung zwischen den beiden Klassen realisiert («GültigkeitExplizit»). Andererseits kann von der Sicht «BergbahnenInGegend» abgeleitet werden, welche Bahnen aufgrund ihrer Lage die Billette eines Tarifbereichs anerkennen.

Auf dieser Basis kann eine Sicht definiert werden, welche die Bergbahnen mit den Billettarten verbindet:

```
VIEW BergbahnUndGueltigeBillettart
  JOIN OF Bb ~ Bergbahn,
         T ~ Tarifbereich,
         Ba ~ Billettart,
         G ~ Gueltigkeit;
  WHERE (G -> Bergbahn == Bb) AND (G -> Tarifbereich == T) AND
        (Ba -> Tarifbereich == T);
=
  BahnNamen: BAG {1..*} OF Bezeichnung := Bb -> Namen;
  BillettNamen: BAG {1..*} OF Bezeichnung := Ba -> Namen;
  Preis: 0.00 .. 5000.00 [Ahland.Taler] := Ba -> Preis;
  Gueltigkeitsdauer: Zeitdauer := Ba -> Gueltigkeitsdauer;
END BergbahnUndGueltigeBillettart;
```

Diese Verbindung kombiniert Bergbahn und Billettart. Sie berücksichtigt dabei die Gültigkeitsbeziehung und dass einer Billettart ein Tarifbereich zugeordnet ist, der mit jenem der Gültigkeits-Beziehung übereinstimmen muss. Damit ist das Ziel schon fast erreicht. Die zulässigen Kombinationen von Bergbahn und Billettart sind als Sichtobjekte verfügbar. Nur möchte man sie noch pro Bergbahn zusammenfassen:

```
VIEW AufBergbahnGueltigeBillettart
  AGGREGATION OF BuGB ~ BergbahnUndGueltigeBillettart
              EQUAL (BuGB -> Bb);
=
  BahnNamen: BAG {1..*} OF Bezeichnung := BuGB -> Bb -> Namen;
  Billettarten: BAG OF BergbahnUndGueltigeBillettart := AGGREGATES;
END AufBergbahnGueltigeBillettart;
```

Eine solche Zusammenfassung erfolgt mit einer Aggregation. Damit werden alle Objekte der Basissicht, die eine bestimmte Bedingung erfüllen (nämlich dass sie zur gleichen Bergbahn gehören), zu einem Sichtobjekt zusammengefasst. Die Menge aller ursprünglichen Sichtobjekte, die zu einem Ganzen zusammengefasst wurde, steht dabei für Strukturattribute zur Verfügung (AGGREGATES).

6.17.4 Sichten erben

Bereits der nationale Verband hat die Sicht definiert, die alle gültigen Billettarten für jede Bergbahn aufführt (Sicht «AufBergbahnGueltigeBillettart», siehe oben). Die Ilistaler wollen diese Sicht auch benutzen. Sie möchten aber auch in dieser Sicht das Attribut Trasseeverlauf einbeziehen, das sie in der eigenen Erweiterung der Klasse Bergbahn definiert haben.

```
VIEW IhBBergbahnUndGueltigeBillettart
EXTENDS BergbahnUndGueltigeBillettart
  BASE Bb EXTENDED BY IhBBb ~ IhBBergbahn
=
  Trasseeverlauf := IhBBb -> Trasseeverlauf;
END IhBBergbahnUndGueltigeBillettart;
```

Mit der Definition einer zusätzlichen Basis (muss eine Erweiterung einer bisherigen Basis sein) stehen deren Attribute zur Verfügung. Beruht ein Sichtobjekt nicht auf dieser Erweiterung (ist es also keine IhBBergbahn), ist das Attribut undefiniert.

- Eine Erweiterung einer Sicht ermöglicht es, Erweiterung der Klassen der Basis-sicht zur Kenntnis zu nehmen und damit deren Attribute auszunützen. Das Bildungsgesetz der Sicht kann damit aber nicht grundlegend verändert werden. Es ist nur möglich, zusätzliche Selektionen zu definieren.

6.18 Namen sind Schall und Rauch – Schema in einer Fremdsprache

Eine Bergbahn ist auf Französisch nicht anders als auf Deutsch. Sie besitzt dieselben Eigenschaften, sie steht mit denselben anderen Klassen in Beziehung, usw. Man mag sich zwar darüber streiten, ob wirklich alles in jeder Sprache gleich gut gesagt werden kann, aber für ein Datenmodell gilt: Die eigentlichen Konzepte sind in jeder Sprache dieselben. Was von einer Sprache zur nächsten wechselt, sind nur ihre Namen.

Wer ein Modell in eine Fremdsprache übersetzen möchte, braucht daher nur alle Bezeichnungen und Anmerkungen auszutauschen. Die eigentliche Struktur – im UML-Diagramm also die Kästchen und Linien – bleibt unverändert. Für INTERLIS-Beschreibungen gibt es ein Werkzeug (den so genannten INTERLIS-Compiler), das prüft, ob eine Übersetzung tatsächlich nur andere Namen verwendet, oder ob beim Übersetzen versehentlich auch die Struktur des Modells gewechselt hat.

7. Die Ilistaler Systeme im Blickfeld

7.1 Was sind normgerechte Systeme? – Systemneutralität

Verschiedene Programmpakete, so das NatTourSys des nationalen Verbandes, aber auch das von den Ilishorn-Bahnen verwendete LiftSys, haben die mittels INTERLIS 2 definierten Modelle implementiert. Sie sind in der Lage, entsprechende Dateien zu erstellen und zu lesen. Sind das nun INTERLIS-Systeme? Allgemein gefragt: Wann erfüllt ein System eine bestimmte Norm?

Eine Beschreibung auf konzeptuellem Niveau hat nicht das Ziel, vorzuschreiben, wie ein Computersystem die beschriebene Vorstellung implementiert. In der Regel werden die Systeme ihre eigenen Möglichkeiten besitzen, um konkrete Anwendungen zu realisieren. Im Idealfall kann das interne Datenmodell direkt aus den Beschreibungen abgeleitet werden. Dies ist jedoch keine Voraussetzung. Es ist durchaus zulässig, dass ein System die Beschreibungen nicht direkt verarbeiten kann. Um kompatibel zu sein, genügt es, wenn ein System direkt oder indirekt (z.B. durch zusätzliche Konversionsprogramme) in der Lage ist, die Daten gemäss der Norm zu behandeln. Eine Norm zur Datenmodellierung sollte darauf verzichten, vorzuschreiben, wie die Systeme eine Anwendung konkret umsetzen. Hier ist Kreativität gefragt, hier soll der Markt spielen. Die Systemunabhängigkeit der Norm ist die Voraussetzung dafür, dass gewünschte Systemwechsel nicht an der Unverträglichkeit der Daten scheitern.

Ein System muss auch nicht die ganze Palette von Fähigkeiten haben, die man sich im Zusammenhang mit einer Beschreibung vorstellen kann. So ist das WebSys z.B. nur in der Lage, INTERLIS 2-Daten einzulesen. Noch extremer ist es mit den Systemen, welche die Zustandsmeldungen von den einzelnen Bergbahnen zur Zentrale schicken. Auf einfachste Art werden Meldungsdaten in vorbereitete Dateien eingefügt, welche gemäss den Regeln der Norm aufgebaut sind.

Stellt eine Anwendung sehr spezielle Anforderungen, kann nicht erwartet werden, dass sie von einer allgemein anwendbaren Norm abgedeckt werden. Die Unterstützung für solche Fälle muss vertraglich mit den Systemherstellern geregelt werden. Allerdings ist dabei Zurückhaltung angebracht, verliert man doch damit einen Teil der Systemunabhängigkeit und schränkt den Kreis der möglichen Systemanbieter ein.

Nachfolgend wird anhand von zwei Beispielen aufgezeigt, wie systemabhängige Aspekte normiert werden können, ohne dass dafür die Interna der Systeme zu stark beeinflusst werden.

7.2 Der Eurokurs schwankt von Tag zu Tag – Parameter und Funktionen

Für jede Billettart ist der Preis in der Landeswährung bekannt. Im Interesse der ausländischen Touristen soll er auch in Euro und US-Dollar bekannt gegeben werden. Man möchte aber nicht bei jeder Änderung der entsprechenden Währungskurse die Preise für alle Billettarten wieder neu eingeben. Sie sollen vielmehr aus der Landeswährung berechnet werden.

```
CLASS Billettart =
  Name: TEXT*100;
  Preis_TALER: 0.00 .. 5000.00 [Ahland.Taler];
  Preis_EUR: 0 .. 4000 [EUR] := TALERtoEUR (Preis_TALER);
  Preis_USD: 0 .. 4000 [USD] := TALERtoUSD (Preis_TALER);
END Billettart;
```

Immer wenn die Daten im System präsentiert oder als Daten für das WebSys bereitgestellt werden, werden die Euro- und Dollarpreise neu aus den Preisen in der Landeswährung berechnet. Die Berechnung erfolgt durch die beiden Funktionen TALERtoEUR bzw. TALERtoUSD. Aber woher soll das LiftSys im Speziellen, aber auch irgendein anderes System wissen, was diese Funktion leisten soll?

Die wichtigste Information besteht aus konzeptueller Sicht in der Tatsache, dass die beiden Fremdwährungspreise nicht als eigenständige Daten behandelt, sondern aus dem Talerpreis abgeleitet werden. Dafür reicht es aus, für eine bestimmte Funktion den Namen und ihre Parameter festzulegen.

```
FUNCTION TALERtoEUR (Taler: NUMERIC [Ahland.Taler]): NUMERIC [EUR]
  // Umrechnung in Euro //;
```

Die eigentliche Leistung der Funktion wird nur als Erläuterung zwischen // angegeben. Die Implementation bleibt Sache der Systeme. Solche Definitionen sollen auf wenige Basismodelle beschränkt sein, denn sie müssen mit den Herstellern der Systeme vereinbart werden. In INTERLIS 2 wird das Bestehen einer solchen Vereinbarung mit einem Kontrakt angemerkt. Funktionen können nur in solchen Modellen definiert werden, für die ein Kontrakt besteht.

```
CONTRACTED MODEL IlisTour AT http://www.interlis.ch/models/ahland
  VERSION "2008-01" =
  ...
END IlisTour;
```

Will man die Anzahl der Funktionen möglichst gering halten, müssen sie möglichst allgemein formuliert werden. Zum Beispiel könnte statt TALERtoEUR eine allgemein verwendbare Funktion für die Division definiert werden.

```
FUNCTION Division (Dividend: NUMERIC, Divisor: NUMERIC): NUMERIC;
```

Allerdings vergibt man sich dabei auch wieder Kontrollmöglichkeiten, denn es besteht keine Gewissheit, dass die erste Angabe ein Betrag in der Landeswährung ist. Mit der nachfolgenden Form ist dies hingegen gewährleistet.

```

FUNCTION ToCurrency (Taler: NUMERIC [Ahland.Taler],
                    Wechselkurs: NUMERIC [Ahland.Taler]):
    NUMERIC [MONEY];

```

Aber woher kommt der Wechselkurs? Man könnte für solche Fälle vorsehen, dass in den Systemen gewisse Werte wie der Euro- oder Dollarkurs als Parameter zur Verfügung stehen. Da auch damit wieder in die Systeme eingegriffen wird, ist auch die Definition von Systemparametern nur innerhalb von Kontrakten erlaubt.

```

PARAMETER
    EURKurs: 0.000 .. 5.000 [Ahland.Taler]; !! Preis eines Euro in Talern
    USDKurs: 0.000 .. 5.000 [Ahland.Taler];

CLASS Billettart =
    Name: Text * 100;
    Preis_TALER: 0.00 .. 5000.00 [Ahland.Taler];
    Preis_EUR: 0 .. 4000 [EUR] := ToCurrency (Preis_TALER, PARAMETER EURKurs);
    Preis_USD: 0 .. 4000 [USD] := ToCurrency (Preis_TALER, PARAMETER USDKurs);
END Billettart;

```

7.3 Auf krummen Wegen – Linienformen

Vielleicht kommt man auf die Idee, den Pistenverlauf nicht mittels Geraden und Kreisbogen zu beschreiben. Man möchte stattdessen beispielsweise Klothoide, Splines oder Bézier-Kurven verwenden. INTERLIS 2 bietet solche Kurvenformen nicht direkt an, lässt aber zu, dass neue Formen für die Linienstücke definiert werden.

Eine Linie besteht aus einer geordneten Menge von Linienstücken. Bei diesen handelt es sich um konkrete Erweiterungen der abstrakten Struktur *LineSegment*. Wer neben den vordefinierten Arten von Liniensegmenten (Geradenstücke und Kreisbögen) zusätzliche benutzen möchte, kann *LineSegment* mit einer passenden Struktur erweitern.

Wiederum muss eine solche Definition mit den Herstellern vereinbart werden. Schliesslich müssen die Systeme ja mit diesen Linienformen fertig werden. Insbesondere möchte man, dass die Linien auf dem Bildschirm und auf Papier korrekt dargestellt werden.

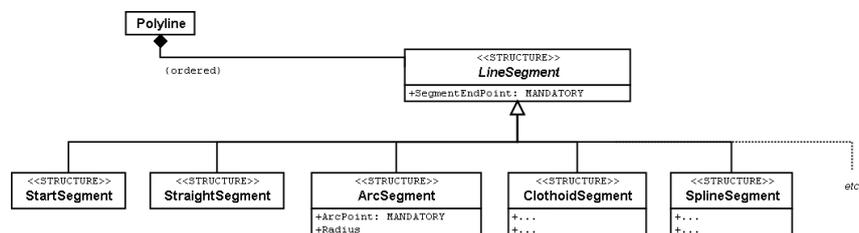


Abbildung 64: INTERLIS-Linien sind aus einzelnen Segmenten zusammengesetzt. Bereits vordefiniert sind Geraden- und Kreisbogenstücke. Die abstrakte Struktur für Liniensegmente kann aber um zusätzliche Formen erweitert werden.

Der Endpunkt eines jeden Segments ist gleichzeitig auch der Anfangspunkt des nächsten. Der Anfangspunkt gehört daher nicht zum Liniensegment. Ein spezielles Startsegment legt fest, wo das erste Liniensegment beginnt.

Ein Kreisbogen wäre mit dem Endpunkt noch nicht genügend bestimmt. Daher besitzen Kreisbögen neben ihrem End- auch einen Hilfspunkt, der ebenfalls auf der Linie liegt. Er sollte sich ungefähr in der Mitte zwischen Anfang und Ende des Segments befinden, weil dann die Berechnung genauer wird.

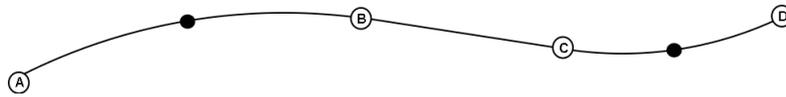


Abbildung 65: Diese Linie besteht aus vier Segmenten: Dem Startsegment mit Endpunkt A, einem Kreisbogen-Segment mit Endpunkt B, einem geraden Segment mit Endpunkt C, und einem weiteren Kreisbogen-Segment mit Endpunkt D. Die Hilfspunkte der beiden Kreisbögen liegen auf der Kurve und sind schwarz gezeichnet.

Der Radius eines Kreisbogens kann selbstverständlich immer aus den Koordinaten der Stützpunkte berechnet werden. Allerdings können rechnerische Ungenauigkeiten dazu führen, dass der errechnete Wert vom beabsichtigten abweicht. Wenn der Radius für die Anwendung eine konzeptuelle Bedeutung besitzt, ist dies nicht akzeptabel. Daher können Kreisbogen-Segmente optional mit einem Wert für den Radius versehen werden.

Wenn der Radius angegeben ist, wird die exakte Lage der Linie mit diesem Wert bestimmt. Der Hilfspunkt dient in diesem Fall nur noch dazu, eine der vier möglichen Verbindungslinien auszuwählen.

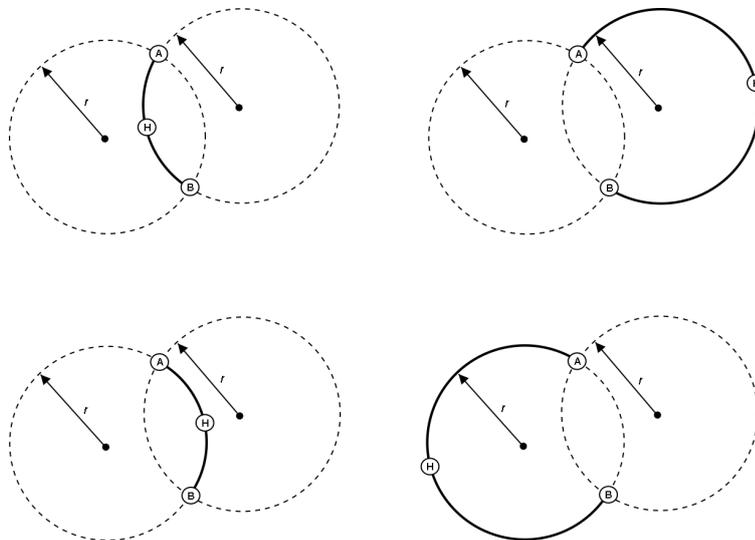


Abbildung 66: Wenn der Radius r angegeben ist, dient der Hilfspunkt H nur noch dazu, einen unter den vier möglichen Kreisbögen auszuwählen, welche die Punkte A und B verbinden.

8. Die Ilistaler Daten unterwegs

8.1 Aus den Augen, aus dem Sinn – Volltransfer

Erinnern wir uns: Die Ilistaler hatten sorgfältig überlegt, welche Daten sie für ihre geplante Anwendung benötigen und ihre Vorstellungen mit einem Datenmodell dokumentiert. In den Gesprächen hatte es sich als nützlich erwiesen, sowohl mit einer graphischen Übersicht als auch mit einer detaillierten Beschreibung in Textform zu arbeiten.

Nach all den langen Diskussionen waren die Ilistaler nun so weit: Das Standard-Programm-paket «LiftSys» wurde beschafft und anhand des Ilistaler Datenmodells eingerichtet. Zwar klappte es doch nicht ganz so auf Anhieb, wie der LiftSys-Verkäufer behauptet hatte, aber die Schwierigkeiten liessen sich rasch beseitigen. Schliesslich konnten die Bahnen, Bahngesellschaften, Skipisten usw. des Ilistals erfasst werden.

Stolz führte der Informatiker vor, wie sämtliche Ilistaler Daten mit einem einzigen Mausklick in eine INTERLIS 2-Datei exportiert werden konnten. Manche im Publikum konnten sich jedoch nicht so schnell begeistern. Es wurden kritische Fragen gestellt: Für diese eine Datei hatte man nun so viele Sitzungen ausgestanden? War das wirklich nötig gewesen?

Der Einwand war allerdings schnell entkräftet, als der Buchhalter vorrechnete, mit wie vielen Talern der Aufwand für den Datenaustausch bisher jährlich zu Buche geschlagen hatte. Jedes Mal, wenn man dem nationalen Tourismusverband Daten geschickt hatte, hatte es Ärger gegeben. Wie viele Arbeitsstunden waren da verloren gegangen!

Neu war es also möglich, dem nationalen Verband eine Datei mit sämtlichen Daten aus dem Ilistal schicken, und dieser konnte sie problemlos weiterverarbeiten.

- Die einfachste Transferart ist der **Volltransfer**, bei dem alle Daten vollständig übermittelt werden.

Dass der Transfer auch dann reibungslos abläuft, wenn die Beteiligten nicht genau dasselbe Datenmodell benutzen, hängt mit der Vererbung zusammen, die in Abschnitt 5.5 erläutert wird.

8.2 Klammern so spitz wie das Ilishorn – Transferregeln auf Basis XML

Was war denn überhaupt in dieser Datei enthalten? Man öffnete sie mit einem normalen Text-Editor.

```
<?xml version="1.0" encoding="utf-8"?>
...
<BASKET BID="xAHTOUIHB01234567" TOPICS="IlistTour.Bergbahnen">
  <IlistTour.IhBBergbahnen.IhBBergbahn TID="xAHTOUIHB04231336">
    <Namen>
      <NatTour.Bezeichnung>
        <Name>Ponylift Ilistdorf</Name>
      </NatTour.Bezeichnung>
    </IlistTour.IhBBergbahn>
  </BASKET>
</IlistTour>
```

```

    <Sprache>de</Sprache>
  </NatTour.Bezeichnung>
</Namen>
...
<LageBergstation>
  <P>
    <C1>8020.60</C1>
    <C2>13188.62</C2>
    <C3>1789.04</C3>
  </P>
</LageBergstation>
<Fahrzeit>
  <Ahland.ZeitdauerInMinuten>
    <Dauer>3</Dauer>
  </Ahland.ZeitdauerInMinuten>
</Fahrzeit>
<Art>Skilift</Art>
...
<BildBergstation>
  http://www.ilishornbahnen.com/webcam?bahn=pony4
</BildBergstation>
...
</IlisTour.IhBBergbahnen.IhBBergbahn>
</BASKET>
...

```

Abbildung 67: Der Ilistaler Ponylift als Teil des Volltransfers für die Ilistaler Daten. Der INTERLIS-Standard legt fest, wie genau eine solche Transferdatei für ein bestimmtes Datenmodell aufgebaut sein muss.

Ziemlich viele spitze Klammern, ganz am Anfang etwas von XML (davon hatte man doch schon mal gehört!), wenn auch in Fragezeichen... Aha, hier war einmal etwas Einfacheres: Der Ponylift Ilistdorf (eher zu Beginn; hier fett gedruckt).

Je nach Datenmodell ist die Transferdatei anders aufgebaut. Die Struktur ist aber natürlich nicht beliebig: Der INTERLIS-Standard schreibt mit präzisen Regeln vor, wie aus einem bestimmten Datenmodell das zugehörige XML-Format abzuleiten ist.

Etwas kompliziert, das Ganze – aber man konnte ja das LiftSys verwenden und musste daher all die Klammern gar nie von Hand eingeben. Immerhin war aber zum Beispiel auch ohne Spezialprogramm ersichtlich, dass die Fahrzeit mit dem Ponylift 3 Minuten dauert.

Der Informatiker erklärte, dass man mit diesen Daten auch noch in vierzig Jahren etwas anfangen können sollte. Auch dann noch, falls die Hersteller der ursprünglichen Programme bis dahin längstens Konkurs gegangen sein sollten, fünfzehn Mal eine neue Programmversion hätte installiert werden müssen, und auch wenn die Geräte bis dahin ganz anders aufgebaut wären. Die Behauptung erschien deswegen glaubhaft, weil es sogar ohne Dokumentation und rein auf dem Papier möglich (wenn auch mühselig) war, die Daten zu verstehen.

Aber was ist dieser komische Text «TID="xAHTOUIHB04231336"»? Er bezeichnet den Ponylift innerhalb der Transferdatei. Der Wert spielt nur für den Transfer eine Rolle: Diese Objekt-Identifikation steht überall dort, wo in der Transferdatei Bezug auf den Ponylift genommen wird. Beim Einrichten von LiftSys konnte angegeben werden, dass alle

Identifikationen mit xAHTOUIHB beginnen sollen, ganz wie es der nationale Tourismusverband in seinem Schreiben gewünscht hatte.

Objektidentifikationen beginnen mit einem Ländercode gemäss ISO 3166 (ch für die Schweiz, fr für Frankreich, etc.) und sechs weiteren Zeichen, die von einer zuständigen, zentralen Stelle eindeutig vergeben werden. Die restlichen acht Zeichen vergibt das Computersystem automatisch.

Anhang D des INTERLIS-Referenzhandbuchs erklärt die Vergabe von Objektidentifikationen.

8.3 Einmal und immer wieder – Inkrementelle Nachlieferung

Ganze siebzehn Minuten auf dem Ponylift? Die armen Kinder sind da doch schon halb erfroren, bevor sie nur schon ihren ersten Stemmbogen gemacht haben! Eine derartige Bahn wird kaum eine Saison überleben; in vierzig Jahren interessiert sich für diese Daten bestimmt niemand mehr.

Schliesslich stellte sich heraus, dass der Ponylift doch nicht gigantisch überdimensioniert worden ist. Es hatte sich einfach jemand beim Erfassen vertippt: Nicht siebzehn, sondern nur eine einzige Minute dauert die Fahrt.

Natürlich war die Zahl schnell im LiftSys geändert. Musste man nun dem nationalen Verband die gesamten Daten gleich nochmals schicken?

- Dank der **inkrementellen Nachlieferung** müssen nach einer Änderung nicht alle Daten, sondern nur die Änderungen übermittelt werden. So müssen weniger Daten verschickt werden. Der Hauptnutzen liegt aber in der Dokumentation, was genau verändert wurde.

Mit dem LiftSys-Programm wurde eine Nachlieferung für den nationalen Verband erzeugt. Im Gegensatz zum Volltransfer enthielt die inkrementelle Nachlieferung nicht die Daten des gesamten Ilistals, sondern nur die Daten von veränderten Objekten. Damit war sie viel kleiner und entsprechend leichter zu handhaben.

Nun macht es nicht immer Sinn, Daten inkrementell nachzuliefern. Der nationale Tourismusverband bietet z.B. den Internet-Dienst, mit dem sich Touristen über die Abonnementspreise bei den verschiedenen Bergbahnen informieren können, nicht selbst an, sondern hat diese Aufgabe an eine Partnerfirma delegiert. Diese erhält dafür periodisch eine vollständige Kopie der Daten im INTERLIS 2-Format. Diese Daten werden durch ein Serverprogramm geladen, das die konkreten Anfragen beantwortet. Eine inkrementelle Nachlieferung würde nur die Anforderungen an dieses Serverprogramm steigern, ohne dass der Nutzen ersichtlich wäre.

8.4 Auch die Blauberge sind touristisch – Behälter, Replikat, polymorphes Lesen

Der nationale Tourismusverband erhält Daten nicht nur aus dem Ilistal, sondern von insgesamt 163 Regionen. Er möchte sich nicht darum kümmern müssen, wie jede einzelne Region ihre Daten pflegen will. Es reicht ihm, hin und wieder aus den Regionen eine Nachlieferung mit den jeweils aktuellen Daten zu erhalten.

In den Regionen werden die Daten in den Datenbanken der verwendeten Systeme gehalten. Im Rahmen von INTERLIS geht man dabei von der Vorstellung aus, dass die Daten jedes

Themas des Datenmodells in einem (oder auch mehreren) Datenbehältern gespeichert sind. So liegen die Bergbahndaten der Ilishornbahnen in einem Behälter, diejenigen der Blaubergbahnen in einem anderen Behälter. Werden die Daten nun von den Ilishornbahnen oder den Blaubergbahnen an den Nationalen Tourismusverband geschickt, ist der jeweilige Behälter auch in der Transferdatei ersichtlich. Das Computersystem des Nationalen Verbandes (NatTourSys) liest die Daten ein und bringt die Datenbank NatTourDB auf den aktuellen Stand. Dabei kann festgehalten werden, woher die Objekte kommen.

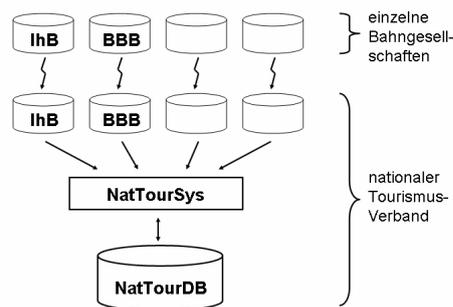


Abbildung 68: Der nationale Tourismusverband erhält von den Ilishorn-, den Blauberg- und vielen anderen Bahnen hin und wieder eine Nachlieferung der jeweiligen Tourismusdaten.

Damit sind nun die Daten zum Ilistaler Ponylift gleich doppelt vorhanden: Einmal bei den Ilishornbahnen, ein zweites Mal beim nationalen Tourismusverband. Das heisst jetzt natürlich nicht, dass die Kinder im Ilistal auf einer neuen Piste snöben könnten. Es wurden ja nur Daten kopiert, keine neuen Lifte gebaut!

Auch elektronisch sind die Verhältnisse klar; die beiden Datenobjekte tragen nämlich die gleiche Objektidentifikation. Hieraus ist ersichtlich, dass es sich um Replikate handelt, die für ein und denselben real existierenden Ponylift stehen.

Mit Replikat verwandte Begriffe sind: Stellvertreter, Duplikate, Proxyobjekte.

Es ist wichtig, dass die Objekt-Identifikation (wie «xAHTOUIHB04231336» im Beispiel oben) wirklich eindeutig ist. Sonst könnten ja zufälligerweise die Ilishorn- und die Blaubergbahnen für zwei verschiedene Objekte dieselbe Identifikation verwenden. Für den nationalen Tourismusverband wäre dann bei einer inkrementellen Nachlieferung nicht klar, ob sich ein Objekt aus dem Ilistal oder eines aus den Blaubergen verändert hat.

Eine Verwaltungsstelle von Ahland («AH») hatte dem nationalen Tourismusverband die Kennzeichnung «AHTOU» zugewiesen. Darum legte der nationale Tourismusverband für jede Bergbahngesellschaft den ersten Teil fest, den sie bei ihren Identifikationen anwenden muss (z.B. «AHTOUIHB» für die Ilishornbahnen und «AHTOU BBB» für die Blaubergbahnen). Für den restlichen Teil der Identifikation ist dann die Bergbahngesellschaft bzw. das von ihr eingesetzte Programm verantwortlich.

Bei einem Volltransfer haben die Objektidentifikationen nicht die gleiche Bedeutung wie bei einer inkrementellen Nachlieferung: Sie müssen nicht erhalten werden, sondern dienen nur

dazu, Beziehungen zwischen den verschiedenen Objekten (z.B. zwischen Tarifbereichen und Billettarten) wiederherzustellen.

8.5 Der Ponylift im «Val de la marmotte jaune» – Fremdsprachen beim Datentransfer

Gleich hinter dem Schwarzen Zahn liegt das «Val de la marmotte jaune». Sieht man einmal davon ab, dass hier französisch gesprochen wird, und dass die lokalen Murmeltiere ein ganz besonders intensiv gefärbtes Fell besitzen, lässt es sich kaum vom Ilistal unterscheiden.

Insbesondere erfreut auch hier ein Ponylift die Kinderherzen. Wie erfährt nun aber der nationale Tourismusverband, wie lange die Fahrt damit dauert? Schliesslich schlagen sich die im Datenmodell benutzten Bezeichnungen auch im Aufbau der Transferdateien nieder. Auf diese Weise kommt es in den Ilistaler Daten zu Zeilen wie `<Dauer>3</Dauer>`. Übersetzt man das Datenmodell in eine andere Sprache, ändert sich damit auch das entsprechende Transferformat.

Wie geht also der nationale Tourismusverband zum Beispiel damit um, dass die Transferdatei aus dem einen Tal die Zeile `<Dauer>3</Dauer>` enthält, jene aus dem Nachbartal aber `<LapsTemps>3</LapsTemps>`?

Der Verband muss nun nicht für jede Landessprache eine separate Software einkaufen. INTERLIS sorgt dafür, dass trotz der Mehrsprachigkeit ein reibungsloser Transfer gewährleistet ist. Die einzige Bedingung ist, dass sich das Datenmodell beim Übersetzen nicht in seiner Struktur verändert hat. Wie in Abschnitt 6.18 angesprochen wurde, steht ein Werkzeug (der so genannte INTERLIS-Compiler) zur Verfügung, mit dem ein übersetztes Datenmodell daraufhin überprüft werden kann, ob es strukturell gleich wie das Original ist.

9. Datenmodellierung über das Ilistal hinaus

Nachdem in den vorangehenden Kapiteln viele Einzelfragen der Datenmodellierung beleuchtet worden sind, greift dieses Kapitel einige grundsätzliche Aspekte auf. Die Lektüre dürfte daher auch für jene von Interesse sein, die sich nicht mit den Details befassen wollen.

9.1 Viele Wege führen aufs Ilishorn – Das richtige Modell gibt es nicht

Die Ilistaler sind sich natürlich bewusst, dass ihr Datenmodell nicht jeder Kritik standhält. Da es das Ilistal in der Realität gar nicht gibt, wurde auch darauf verzichtet, das Modell unter verschiedenen Gesichtspunkten immer wieder zu hinterfragen. Schliesslich ging es primär darum, die verschiedenen Aspekte der Modellierung und die entsprechenden Instrumente vorzustellen.

Bei jedem Modell stellt sich aber die Frage: Ist das Modell richtig? Oder fragt man vielleicht besser: Ist das Modell gut?

Sicher greift die Haltung zu kurz, die jedes Modell als richtig oder gut bezeichnet, das die Anforderungen der Beschreibungssprache rein formal erfüllt. Nur weil ein Modell vom INTERLIS-Compiler akzeptiert wird, ist es sachlich noch nicht gut.

Die Haltung entsprechend dem Sprichwort «Viele Wege führen nach Rom» – oder eben aufs Ilishorn – kommt der Sache schon näher. Es ist offensichtlich, dass es verschiedene Möglichkeiten gibt, einen bestimmten Sachverhalt zu modellieren. Nicht alle möglichen Modelle sind aber gleich gut. Schliesslich gibt es auch breite, schmale, holprige, steile, befahrbare, aussichtsreiche Wege aufs Ilishorn. Nicht jeden Tag wählt man den gleichen Weg als den besten aus. Die Bedürfnisse und damit die Beurteilungskriterien sind schliesslich auch nicht immer die gleichen.

In den nächsten Abschnitten werden deshalb verschiedene Beurteilungskriterien beleuchtet, die für die Datenmodellierung von einiger Bedeutung sind.

9.2 Das Ilishorn aus verschiedenen Blickwinkeln – Originaldaten und Sichten

Orientiert man sich bei der Datenmodellierung am unmittelbaren Bedürfnis, besteht die Gefahr, dass die Daten ungeeignet erfasst werden. So ist es z.B. ungeeignet, bei einer Person das Alter als Attribut vorzusehen. Dieses ändert schliesslich dauernd. Besser ist es, das Geburtsdatum als Attribut festzulegen und das Alter mittels einer Sicht aus dem Geburtsdatum und dem aktuellen Datum abzuleiten (vgl. Abschnitt 6.17.2).

Wenn das Datenmodell so gebildet ist, dass es der eigentlichen Sache und nicht einer bestimmten Betrachtungsweise entspricht, können wir für die konkreten Nutzungen flexibel reagieren und sie aus den Originaldaten ableiten. Die Daten enthalten dann eben das

Ilshorn und nicht den Blick von Ilisdorf oder Ilisbad aus. Diese und beliebige andere Perspektiven können als Sichten aus den Daten abgeleitet werden.

Manchmal ist die Ableitung der gewünschten Sichten aus «ideal» modellierten Daten aber recht anspruchsvoll. Insbesondere Sichten, die Bildungsgesetze mit geometrischen Berechnungen enthalten, sind nicht immer einfach realisierbar. Der Entscheid, wann von der Tugend abgewichen werden kann und das Datenmodell an der Sicht ausgerichtet wird, ist nicht immer einfach und hängt in der Regel von weiteren Beurteilungskriterien ab. Auch wenn man sich für ein Modell entscheidet, das nicht die eigentlichen Daten sondern eine Sicht auf sie beschreibt, lohnt es sich aber, zu überlegen, welches das «schöne» Modell wäre. Es fällt dann leichter, zu gegebener Zeit auf dieses Modell zurückzukommen und damit die Nachteile des sichtenorientierten Modells zu überwinden.

Der Hauptnachteil eines sichtenorientierten Modells zeigt sich in der Erfassung und vor allem in der Nachführung der Daten. Führt man z.B. für jeden Gebäudeeingang den Strassennamen, die Hausnummer, die Postleitzahl und die Ortschaft als direktes Attribut, besteht die Gefahr, dass sich Fehler, z.B. unterschiedliche Schreibweisen, einschleichen. Zudem ist der Erfassungsaufwand wesentlich grösser. Noch schlimmer wird es in der Nachführung, wenn z.B. die neue Siedlung Ilismühle als Ortschaft mit eigener Postleitzahl eingeführt und der hintere Teil von Ilisdorf dieser neuen Ortschaft zugeordnet wird. Alle betroffenen Gebäudeeingänge mutieren? Besser wäre es, man könnte einfach die neue Ortschaft samt ihrem Gültigkeitsbereich einführen und den Gültigkeitsbereich von Ilisdorf entsprechend anpassen.

Mit guten Modellen erreicht man, dass ein veränderter Sachverhalt in der Realität zu genau einer Änderung in den Daten führt. Alles Weitere kann mittels Sichten abgeleitet werden.

9.3 Nagt der Zahn der Zeit am Ilshorn? – Lebenszyklen

Gute, d.h. sachgerechte, von der unmittelbaren Nutzung unabhängige Datenmodelle sind um so wichtiger, je langlebiger und allgemein brauchbarer die mit den Daten beschriebenen Sachverhalte sind. Bei Daten, die für einen privaten Zweck oder für ein zeitlich klar begrenztes Projekt, das mit einem bestimmten System abgewickelt wird, erhoben werden, ist das Datenmodell nicht so wichtig wie für Daten, die langlebig sind.

Daten, die die reale Welt beschreiben, sind aber in der Regel langlebig. Das Ilshorn selbst verändert sich nur in geologischen Zeiträumen. Aber selbst Häuser, Bahnen, Rechtsverhältnisse haben Lebensdauern von einigen Jahren bzw. Jahrzehnten. Die Objekte in der Realität werden darum die Computersysteme, auf denen ihre Abbilder beschrieben sind, überdauern.

Damit auch die Abbilder der Realität, also die Datenobjekte die spezifischen Computersysteme (Hardware und Software) überdauern können, braucht es eine klare Vorstellung über die Datenobjekte. Es braucht ein Datenmodell.

Dieses Datenmodell soll möglichst systemunabhängig sein. Denn das nächste System hat voraussichtlich andere Fähigkeiten. Vielleicht können dann Sichten automatisch gebildet werden. In Anbetracht der relativ kurzen Lebensdauer von Softwarepaketen sollte man sich beim Strukturieren der Daten nicht zu stark davon leiten lassen, welche Funktionalität ein bestimmtes Softwarepaket zurzeit besitzt.

9.4 Vor lauter Bäumen den Wald nicht mehr sehen – Detaillierungsgrad

Es steht ausser Frage, dass die reale Welt nicht so modelliert wird, dass die einzelnen Atome, aus denen die Realweltdinge bestehen, die Datenobjekte sind. Das wäre viel zu mühsam. Ein Gebäude soll sich auch nicht aus den einzelnen Backsteinen und Ziegeln zusammensetzen. Aber wann sprechen wir von einzelnen Bäumen, wann von einem Wald, ohne die einzelnen Bäume als Datenobjekte aufzuführen?

Natürlich hängt dies davon ab, was mit dem System erreicht werden soll. Wichtig dabei ist, dass es nicht nur darum geht, ein möglichst gutes Modell zu finden. Dieses muss auch umsetzbar sein. Ist das Modell verständlich? Ist es möglich, für die Datenerfassung und die Nachführung einen Dialog zu definieren, den die Menschen, die für die Arbeit vorgesehen sind, ohne Probleme bewältigen können?

Ein einfaches Modell, dessen verschiedene Objektklassen nicht allzu sehr miteinander verflochten sind, kann sich gerade in dieser Hinsicht positiv bemerkbar machen.

Je komplizierter ein Modell ist, desto wichtiger ist es, dass man dennoch eine Bedienungsfläche findet, die intuitiv verständlich ist. In solchen Situationen ist es gefährlich, wenn ein Grundmodell so erweitert wird, dass nicht nur einzelne Attribute angefügt, sondern neue Beziehungen definiert werden. Die Gefahr, dass ein für das Grundmodell entworfener Bedienungsdialg seine Eleganz und Verständlichkeit verliert, ist gross.

Einmal mehr gilt: «So einfach wie möglich, so komplex wie nötig». Die gute Umsetzung dieses Grundsatzes ist aber bekanntlich nicht einfach.

9.5 Das Kurorchester Iisbad spielt auf – Datenmodellierung ist anspruchsvoll

Wenn sich Pauken und Trompeten, Geigen und Flöten zu einem Ohrenschmaus vereinigen, vergessen wir verständlicherweise, wie viel Aufwand dahinter steckt. Was hat es nur gebraucht, bis alle nach ersten Blockflötenversuchen in den Kindesjahren so virtuos mit ihrem Instrument umgehen konnten! Aber auch jetzt wird immer wieder geübt, allein und im Orchester.

Es ist zu hoffen, dass bei der Datenmodellierung auch so ernsthaft gearbeitet wird. Wer Modelle verstehen will, muss die verschiedenen Instrumente mindestens passiv kennen. Wer Modelle definieren will, tut gut daran, immer wieder zu üben, bis die Instrumente (Klassen, Attributtypen, Beziehungen, Vererbung, usw.) in ihrer Wirkung klar sind. Auch wenn das Resultat noch so gut erscheint, es lohnt sich, nochmals darüber nachzudenken, das Modell der Kritik anderer Leute auszusetzen und es nochmals aus verschiedenen Richtungen zu beleuchten. Nur so können gute Modelle entstehen, Modelle, die so einfach und verständlich sind, dass sie sich in der Praxis bewähren.

10. Schlussbericht für Behörden und Management

Natürlich hat man das gute Funktionieren der neuen Lösung im Ilistal gebührend gefeiert. Die Gemeindepräsidentin legte aber Wert darauf, über die technischen Dinge doch etwas genauer Bescheid zu wissen. Was hat zum Erfolg beigetragen? Warum konnte man die ganze Aufgabe unter Beibehaltung der bisherigen Systeme lösen? Warum wird die Gefahr als klein eingeschätzt, dass eine grosse Überarbeitung nötig ist, wenn eines der beteiligten Systeme durch etwas Neues ersetzt wird? Zur Vorbereitung ihrer kurzen Ansprache hat sie darum den Bericht gelesen, den der Bausekretär und der technische Leiter der Ilishornbahnen verfasst haben.

* * * * *

I. Erinnerungen an den Startschuss

Noch ist die Startpräsentation, wie man sich eine neue Ilistaler Webpage vorstellen könnte, in bester Erinnerung. Nach einigen imposanten Demonstrationen geriet man sich fast in die Haare. Zum Glück waren verschiedene Fachleute anwesend. Sonst hätten wir Laien gar nicht gemerkt, wie kontrovers die Meinungen unter den Fachleuten sind, wie schlecht Argumente und Gegenargumente – oder besser Schlagworte und Gegenschlagworte – wirklich zusammenpassen. Dank der Diskussion, die fast zu einem Disput wurde, sind wir hellhörig geworden.

II. Sachen, Abbilder und der Zahn der Zeit

Der Beizug eines weiteren Fachmanns hat sich gelohnt. Die ersten Gespräche waren dann zwar etwas überraschend. Technische Schlagworte fehlten oder wurden wieder vom Tisch gefegt. Dafür folgte eine gründliche Auseinandersetzung mit der Sache.

Erste Erkenntnis: Sache ist nicht, was im Computer drinnen steckt oder gar, was er als Bilder, Tabellen und andere Präsentationen ausspuckt, sondern was in der Realität existiert und passiert.

Das, was im Computer drin ist, kann mit einem Holzmodell oder einem Sandkasten verglichen werden. Die Fotos, die wir davon aus verschiedenen Richtungen und mit verschiedener Beleuchtung machen, können mit dem Bildschirm oder mit Ausdrucken verglichen werden. Und eines ist ja auch bei Holzmodellen oder Sandkasten offensichtlich: Die Realität zum ersten Mal abzubilden ist eine Riesenarbeit. Bis da all die Häuser, Strassen, Bahnen, die in der Realität existieren, auch im Abbild in vereinfachter Form aufgebaut sind! Noch schlimmer als die Ersterfassung ist dann die Nachführung: Schon nach kurzer Zeit müssen Änderungen gemacht werden, um das Abbild aktuell zu halten. Und der Zahn der Zeit nagt an den Sand- und Gipsstrukturen. Da hat man es mit dem Computer schon besser!

Natürlich muss auch hier nachgetragen werden, wenn sich die Realität ändert, aber wenigstens veralten die verbleibenden Informationen nicht.

Wirklich? Nein, wirklich nicht so schnell wie im Sandkasten. Aber nach wenigen Jahren möchten wir wieder einen neuen, leistungsfähigeren Computer kaufen, und nach nochmals etwas längerer Zeit ist auch die Software für die Bearbeitung nicht das Beste, was es gibt. Man möchte neue Programme einsetzen. Und wie bringt man jetzt das elektronische Abbild der Realität vom alten System aufs Neue? Wie bringt man quasi den Sand vom alten in den neuen Sandkasten, ohne dass er zwischen den Händen zerrieselt und das Werk neu aufgebaut werden muss?

III. Systemunabhängige Beschreibung der Struktur der Daten

Damit man elektronische Abbilder der Realität – also Daten – von einem System auf ein anderes übertragen kann, braucht es klare Vorstellungen darüber, wie die Daten aussehen, wie sie strukturiert sind. Die Knacknuss dabei: Auch diese Strukturbeschreibungen müssen unabhängig von konkreten Systemen sein. Sie müssen so gemacht sein, dass sie von den Anwendungsfachleuten und von den technischen Fachleuten aller beteiligten Stellen gleich interpretiert und verstanden werden. Im idealen Fall ist die Beschreibung sogar so aufgebaut, dass nicht nur Menschen, sondern auch Computerprogramme etwas damit anfangen können. Es braucht ein formalisiertes Esperanto der Datenbeschreibung.

In der Alltagssprache ist Esperanto ohne Bedeutung geblieben. Das Lebendigere, vielleicht auch das für den Stärkeren Bequemere hat sich durchgesetzt. Im technischen Bereich liegt die Sache aber etwas anders: Wer sich nicht um Systemunabhängigkeit bemüht, wird von Systemen abhängig. Und Systeme sind nun einmal relativ kurzlebig. Daher hat sich die Erkenntnis, dass es ein Esperanto im Sinne von systemunabhängigen Normen braucht, trotz vieler De-facto-Standards etabliert. Mit der *Unified Modeling Language* (UML) hat eine graphische Sprache im Bereich der Datenmodellierung weltweit eine recht grosse Verbreitung gefunden.

In intensiven Gesprächen wurde ein Modell der Realität entworfen. Dabei galt es zunächst zu klären, welches überhaupt die massgeblichen Gegenstände sind, wie man sie am verständlichsten benennt, wie sie zueinander in Beziehung stehen und welche weiteren Eigenschaften sie haben. Das Protokoll dieser Gespräche hielt die Erkenntnisse immer in der UML-Bildersprache fest und ergänzte sie mit Erklärungen und Kommentaren. Da es sich zeigte, dass die Bilder durch viele Details überladen werden und zum Teil auch nicht die nötige Präzision aufweisen konnten, wurde mit INTERLIS noch ein zusätzliches Beschreibungsmittel eingesetzt: Ähnlich wie in einer Programmiersprache können damit die Datenstrukturen präzise beschrieben werden.

Eine wichtige Eigenschaft, die durch UML und INTERLIS 2 unterstützt wird, besteht darin, dass auch Datenbeschreibungen, die anderswo gemacht wurden, genutzt werden können. So konnte man von Festlegungen der amtlichen Vermessung profitieren, zum Beispiel bei den Gebäudeadressen. Die Datenbeschreibung des nationalen Tourismusverbandes wurden nicht nur genutzt, sondern zusätzlich auch entsprechend den lokalen Bedürfnissen ergänzt.

IV. Systemunabhängiger Datentransfer

Ähnlich wie bei der Beschreibung der Struktur der Daten verhält es sich mit den Daten selbst. Es braucht also nicht nur eine Norm für die Datenbeschreibung, sondern auch eine Norm für den Datentransfer. Beim Übertragen der Daten von einem System auf ein anderes wird quasi Esperanto gesprochen, innerhalb der einzelnen Systeme dagegen die jeweilige Muttersprache.

Nachdem man sich so intensiv mit dem Aufbau der Daten auseinandergesetzt hat, möchte man sich eigentlich nicht mehr um diese Details kümmern. Hier hat sich der Einsatz von INTERLIS gleich nochmals gelohnt: Mit der Beschreibung der Datenstruktur ist nämlich auch gleich festgelegt, wie die zugehörigen Daten transferiert werden.

Wichtig ist dabei, dass die Datenbeschreibung nicht nur von Menschen gelesen und verstanden werden kann. Wegen der formalen Sprachdefinition kann sie auch durch Computerprogramme gelesen werden. Aufgrund von Datenbeschreibung und Transferregeln ist klar, wie in der Folge der Datentransfer aufgebaut sein muss.

V. Unterschiedliche Auswirkungen auf die beteiligten Systeme

Im Rahmen der Gesamtlösung wurden verschiedene Computersysteme eingesetzt. Insbesondere konnten die bisherigen Systeme weiterhin genutzt werden. UML und INTERLIS schreiben nämlich nicht vor, wie die Systeme aufgebaut sein sollen, sondern nur, wie die Daten letztlich geliefert werden, damit sie der konzeptuellen Vorstellung entsprechen.

Das System des Bauamtes war direkt in der Lage, die zu transferierenden Daten zu erstellen und empfangene Daten zu lesen. Beim System der Ilishornbahnen wurde ein Konverter eingesetzt, also ein Programm, welches die system-spezifischen Daten in ein neutrales Transferformat umsetzte. Bei den kleinen Erfassungssystemen an den Talstationen konnte mit einfachsten Programmen dafür gesorgt werden, dass auf Grund der Messung jeweils ein korrekter Datensatz als inkrementelle Nachlieferung an die Zentrale gesendet wird.

Interessant ist auch ein Blick auf das System des nationalen Verbandes. Diesem konnten nämlich Dateien geschickt werden, die auch die durch uns definierten Zusätze enthalten. Dank dem Prinzip des polymorphen Lesens störte das überhaupt nicht.

Ein Grund, dass dies alles möglich war, lag wohl auch daran, dass INTERLIS 2 auf dem internationalen Standard der *Extensible Markup Language* (XML) aufbaut.

VI. Ausblick

Auf Grund der gemachten Erfahrungen mit all den beteiligten Systemen sind wir zuversichtlich, dass auch neue Wünsche und neue Systeme in unsere Lösung integriert werden können, ohne das Ganze zu gefährden. Damit können wir ein wichtiges Ziel erreichen: Unsere Lösungen sind keine Eintagsfliegen, sondern nachhaltig.

* * * * *

Die Gemeindepräsidentin flocht die wichtigsten Dinge in ihr Referat ein und erkannte richtig, dass der technische Erfolg nicht zuletzt dank der anfänglichen Distanz zur Technik zustande kam. Und beim Apéro freute sie sich insgeheim ganz besonders: Sie hatte wieder einmal

eine gute Nase gehabt, als sie sich damals bei der ersten Präsentation im Gemeindehaus an das Gespräch mit ihrer Kollegin erinnerte und sich für ein sorgfältiges Vorgehen entschloss.

Register

Verweise auf das ausführliche Beispiel in Kapitel 4 sind in *Schrägschrift* gesetzt.

- A**
- Abstrakt 35, 46
 - Adresse 29, 116
 - auf dem Internet..... *Siehe* URI
 - Aggregation 97
 - bei Beziehungen 35, 79
 - bei Sichten 97
 - Allgemeiner Fall *Siehe* Vererbung
 - Amtliche Vermessung 29, 116
 - Area *Siehe* Gebietseinteilung
 - Arten 35, 53
 - Assoziation 35, 79
 - Attribut 13
 - bei Beziehungen 80
 - fakultativ 57
 - obligatorisch 57
 - Attributtyp *Siehe* Datentyp
 - Aufbau auf Vorhandenem 17
 - Aufschlüsselung 97
 - Aufzählung 35, 54
 - abgeschlossen 56
 - geordnet 40, 55
 - strukturiert 56
 - und Vererbung 37, 54
 - zyklisch 40, 56
 - Ausprägung 12
- B**
- Basisklasse *Siehe* Vererbung
 - Behälter 92, 107
 - Beschreibung *Siehe* Datenmodell
 - Beziehung 11, 35, 78
 - abgeleitet 38, 85
 - Aggregation 79
 - Assoziation 79
 - Beziehungsklasse 36
 - geordnet 82
 - Komposition 39, 79
 - mehrgliedrig 36, 81
 - Stärke 79, 85
 - themenübergreifend 92
 - und Vererbung 83
 - Bézierkurve 103
 - Bezugssystem
 - von Koordinaten 65
 - von Zeitpunkten 77
 - Bildungsgesetz *Siehe* Sicht
 - Booch 27
- C**
- Codierung 24
 - Compiler 30, 99, 109
- D**
- Datenaustausch ... *Siehe* Transfer
 - Datenbeschreibung 27
 - Datenmodell 11, 115
 - Darstellung 27
 - mehrsprachig 99
 - standardisiert 29, 93
 - Datensatz 12
 - Datentyp 14, 32, 58
 - geometrisch 65
 - numerisch 36, 52
 - strukturiert 76
- E**
- Eigenschaft 13
 - abstrakt 53
 - erben *Siehe* Vererbung
 - Eindeutigkeit 34, 88, 108
 - Einheit 32, 52
 - Einschränkung.. *Siehe* Vererbung
 - Ellipsoid 59
 - Ellipsoidhöhe 62
 - Enklave 68
 - Entitätsmenge 11
 - Erben *Siehe* Vererbung
 - Erweiterung *Siehe* Vererbung
 - Exemplar 12
 - Existenzerfordernis 90
- F**
- Feld 13
 - Fläche 67
 - Fließkommazahl 51
 - Föderalismus 28, 47, 117
 - Funktion 37, 102
- G**
- Ganzzahl 51
 - Gebietseinteilung 40, 68
 - Genauigkeit
 - von Linien 66, 104
 - von numerischen Werten 52
 - Geoid 62
 - Geradenstück 66, 103
 - GML 30
 - Grafik 25
 - Größenordnung 51
- H**
- Hilfspunkt 104
 - Höhe 62
- I**
- Identifikation
 - beim Transfer 106, 108
 - mit Attributen 88
 - Vergabe 107
 - Inkrementelle Nachlieferung 16
 - Inspection 97
 - Integer 51
 - Internet-Adresse *Siehe* URI
 - ISO 19100 29
 - ISO 3166 107
- J**
- Join 95

K

Kardinalität	13
Klassendiagramm	11
Klothoide	103
Kolonne	13
Komposition	39, 79
Konsistenzbedingung	38, 86
für Stützpunkte	91
und Vererbung	91
Kontrakt	102
Koordinaten	32, 35, 58
im Raum	32, 62
zum Etablieren von	
Beziehungen	85
Koordinatensystem	59
Kreisbogen	66, 104

L

Leitungsnetz	29
Linie	65
Aufbau	66, 72, 103
gerichtet	67
im Raum	69
speziell geformt	103

M

Mehrfachvererbung	49
Mehrsprachigkeit	72
bei Objekteigenschaften	34
beim Transfer	109
von Datenmodellen	99
Metadaten	60
Modell	<i>Siehe Datenmodell</i>
Modellbasierter Ansatz	30
Modul	17

N

Nachführung	115
Nachhaltigkeit	106
Nachlieferung	16
Numerischer Datentyp	36, 52

O

Oberklasse	<i>Siehe Vererbung</i>
Object Management Group	30
Objekt	12
Objektidentifikation	<i>Siehe Identifikation</i>
Objektkatalog	11
Objektklasse	11
OMG	30
OMT	27
OOSE	27

P

Package	17
Paket	17
Parameter	102
Plausibilität	87
Polymorphes Lesen	24, 107, 117
Polymorphismus	47
beim Transfer	24, 107, 117
Projektion	
bei Sichten	95
von Koordinaten	59
Proxy-Objekt	107

R

Radius	104
Real (Fließkommazahl)	51
Replikat	107
Richtung	64
Rolle	35, 78

S

Schema	<i>Siehe Datenmodell</i>
Schwerehöhe	62
Segment	66
Sicht	25, 38, 93
Aufschlüsselung	97
Bildungsgesetz	94
schrittweiser Aufbau	97
und Vererbung	98
Verbindung	95
Vereinigung	97
Zusammenfassung	97
Spezialfall	<i>Siehe Vererbung</i>
Spline	103
Stellvertreter	107
Struktur	71
als Aufschlüsselung	97
als Zusammenfassung	97
und Vererbung	71
Subklasse	<i>Siehe Vererbung</i>
Superklasse	<i>Siehe Vererbung</i>
Systemneutralität	101

T

Tabelle	11
Text	34, 57
Thema	19, 91
Topic	<i>Siehe Thema</i>
Transfer	16, 105
an übergeordnete Stellen	24
der Beispieldaten	41
Format	22
Identifikation	106
inkrementell	107, 108

mit Konversionsprogramm	24, 101
modellbasiert	22, 117
polymorphes Lesen	107
systemspezifisch	23
systemunabhängig	117
und Mehrsprachigkeit	109
vollständig	16, 108
von Linien	104
Tupel	12
Typ	<i>Siehe Datentyp</i>

Ü

Überlappung	
bei Flächen	<i>Siehe Gebietseinteilung</i>
bei Linien	66

U

Umlaute	24
Undefiniertheit	57
Unified Modeling Language	27, 116
UML-Editor	30
Union	97
Unteraufzählung	56
Unterklasse	<i>Siehe Vererbung</i>
URI	38, 57

V

Verbindung	95
Vereinigung	97
Vererbung	
im Grossen	20, 47
mehrfach	49
Prinzipien	43
übertrieben detailliert	44
von Aufzählungen	37, 55
von Beziehungen	83
von Konsistenzbedingungen	91
von numerischen	
Wertebereichen	52
von Sichten	98
von Strukturen	71
von Themen	20, 47
Verknüpfung	<i>Siehe Beziehung</i>
Vermessung	29, 116
Verträglichkeit	47
Verweis	<i>Siehe Beziehung</i>
View	<i>Siehe Sicht</i>
Volltransfer	16

W

Währung	52, 102
---------------	---------

Wertebereich 14, 32, 58
 geometrisch 65
 numerisch 36, 52
 strukturiert 76
Winkel 64

X

XML 29, 105, 117

Z

Zeichencodierung 24

Zeichenkette 34, 57
Zeichenvorrat 24
Zeiger *Siehe* Beziehung
Zeitdauer 33, 35, 74
Zeitpunkt 33, 77
Zusammenfassung 97