

**Konzept****INTERLIS-Workbench - Phase 1:***Bedarfsabklärung der Funktionalität*

Auftraggeber	Strategie Geoinformation Schweiz/GeoStandards.ch
Dienstleister	KSTEC GmbH – Platz 10 - 6039 Root D4
Projektleitung	Frank Gottsmann, Peter Staub, Stefan Henrich, Kurt Spälti
Autoren	Projektleitung
Klassifizierung	Zur Genehmigung
Status	abgeschlossen

**Änderungsverzeichnis**

Datum	Version	Änderung	Autoren
Juli 2024	0.1	Dokument erstellen; Einleitung, Ausgangslage und Ziele einfügen	Kurt Spälti
August 2024	0.2	Rahmenbedingungen Abgrenzung, Anforderungen einfügen	Kurt Spälti
August 2024	0.3	Input der Projektleitung einfügen	Kurt Spälti
29.08.2024	0.4	Version für die Umfrage bei den Stakeholdern	Projektleitung
05.11.2024	0.5	Umfrageauswertung mit Bedürfnisnachweis einfügen	Projektleitung
11.11.-16.11.2024	0.6	Review; Konzept überarbeiten	Projektleitung
18.11.2024	1.0	Letzte Anpassungen; finale Version	Projektleitung

## Inhaltsverzeichnis

1	Einleitung.....	3
2	Bedürfnisnachweis aufgrund der Umfrage .....	4
2.1	Auswertung .....	4
2.2	Bedürfnisnachweis für eine umfassende INTERLIS-Workbench.....	5
3	Ausgangslage .....	6
3.1	Erkenntnis .....	6
4	Kernziele.....	7
4.1	Rahmenbedingungen .....	7
4.2	Abgrenzung .....	7
5	Anforderungen .....	9
5.1	Editierbare Sichten aufs Datenmodell .....	9
5.2	Auswahl von Objekten und Funktionen.....	13
5.3	Modellübersetzungen .....	14
5.3.1	Sicht auf die Übersetzungstabellen .....	14
5.3.2	Sicht auf das gesamte Modell in einer bestimmten Sprache .....	15
5.4	Erstellen, editieren, übersetzen von Katalogen.....	15
5.5	Modell prüfen, Hilfefunktion .....	16
5.6	Darstellungskatalog.....	16
5.7	Modelldokumentation .....	17
5.8	Modell und Modellbestandteile exportieren.....	18
5.9	Modell oder Modellbestandteile importieren.....	18
5.10	Unterstützung von Validierungsmodellen .....	18
5.11	Technologie und Kollaboration.....	19
5.12	Modelltransformationen.....	19
6	Glossar .....	20

# 1 Einleitung

Gestützt auf das Geoinformationsgesetz (GeoIG) sowie die Geoinformationsverordnung (GeoIV) legt das Bundesamt für Landestopografie (swisstopo) die allgemeine Beschreibungssprache für Geobasisdaten fest. In der Verordnung des Bundesamtes für Landestopografie über Geoinformation (GeoIV-swisstopo) ist INTERLIS als allgemeine Beschreibungssprache für Geodatenmodelle festgehalten.

Die modellbasierte Methode mit dem INTERLIS-Standard und die entsprechenden Werkzeuge sind etabliert, bewährt und vielfach erfolgreich im Einsatz. Die Vorzüge der Methode und gesetzliche Grundlagen (GeoIG, GeoIV) haben dazu geführt, dass die Entwicklungen und Projekte um INTERLIS – oder allgemein: Projekte und Entwicklungen um Geodaten-Infrastrukturen – eine hohe Dynamik erhalten haben.

Allerdings wurden die Weiterentwicklungen sowohl des Standards als auch der Werkzeuge teilweise als unkoordiniert, intransparent und mit zu geringem Fokus auf die Nutzer/Nutzung wahrgenommen. Als Handlungsmassnahme wurde die Organisation GeoStandards.ch gegründet und im Bereich der NGDI mit der Weiterentwicklung von Standards, den zugehörigen Werkzeugen und Weiterbildungsangeboten betraut.

GeoStandards.ch hat zum Ziel die nachhaltige und nutzenorientierte Standardisierung im Bereich Geoinformation in der Schweiz sowie die wirksame Steuerung der Entwicklung von Lösungen und Software-Werkzeugen im Umfeld von GeoIG und nationaler Geodaten-Infrastruktur (NGDI) sicher zu stellen.

In der Strategie GeoStandards.ch ist der allgemeine Rahmen für alle Aktivitäten innerhalb der neuen Organisation GeoStandards.ch beschrieben. Die Strategie definiert, wie sich die Schweizer Geoinformatik-Landschaft inklusive INTERLIS in den nächsten fünf bis zehn Jahren entwickeln soll.

Konform zur Strategie GeoStandards.ch wurde das Projekt «Initialisierung INTERLIS-Suite» realisiert, um die koordinierte Weiterentwicklung von Werkzeugen im Bereich INTERLIS zu gewährleisten. Die erarbeiteten Ergebnisse dienen als Grundlage für Projekte, welche die Neu- oder Weiterentwicklung von konkreten INTERLIS Software-Tools zum Ziel haben.

Das vorliegende Konzept dient als Bedarfsabklärung der Funktionalität für einen umfassenden INTERLIS Modell-Editor – hier mit dem Arbeitstitel «**INTERLIS-Workbench**» bezeichnet. Die Modellierenden und weitere Nutzerinnen und Nutzer sollen sich mittels Umfrage an der Bedarfsabklärung beteiligen. Die Rückmeldungen werden anschliessend in harmonisierter Form in dieses Konzept eingearbeitet.

## 2 Bedürfnisnachweis aufgrund der Umfrage

Die Umfrage beinhaltete elf geschlossenen Fragen, welche entweder mit ja/nein oder aus einer Auswahl von vorgegebenen Antworten beantwortet werden konnten, wobei jeweils auch mehrere Antworten ausgewählt werden konnten. Ergänzend dazu gab es zehn offene Fragen bei denen die TeilnehmerInnen Präzisierungen oder zusätzliche Anforderungen zu einem Thema beschreiben konnten.

53 Personen haben die Personalien ausgefüllt, wobei es pro Frage zwischen 9 und 49 Antworten gab. Naturgemäss wurden die geschlossenen Fragen häufiger, als die zur Ergänzung formulierten offenen Fragen beantwortet.

Grundsätzlich waren die Antworten sehr positiv. Es wurde jedoch – zu Recht – bemängelt, dass die Umfrage nur in Deutsch erfolgte!

### 2.1 Auswertung

Mehr als  $\frac{3}{4}$  der Teilnehmenden benutzen den heutigen UML/INTERLIS-Editor wöchentlich, monatlich oder einige Male im Jahr, wobei die Hälfte der Teilnehmenden das Werkzeug zum Erstellen oder ändern von Datenmodellen benutzen und ein Viertel ausschliesslich um UML-Diagramme oder Objektkatalogentwürfe damit zu erzeugen.

98% sind der Meinung, dass es eine umfassende INTERLIS-Workbench braucht, mit modernerer und intuitiverer User-Experience für das Visualisieren, Diskutieren und Verwalten von INTERLIS-Datenmodellen. In den erweiterten Antworten werden die Aussagen der Ausgangslage, der Konklusion und summarisch der Ziele und Anforderungen des Konzepts bestätigt.

Die formulierten Kernziele erhielten je rund 15% Zustimmung, wobei die Nutzung von KI für die Modellierung nur rund 8% bestätigten. Zusätzlich wird darauf hingewiesen, dass es eine Priorisierung braucht, um möglichst rasche die Kernfunktionen der INTERLIS-Workbench zu erhalten .

#### Bezüglich der Anforderungen:

Die vier editierbaren Sichten aufs Datenmodell (UML-Diagramm, Objektkatalog, INTERLIS-Code und Strukturbaum) werden jeweils etwa gleich häufig als hilfreich erachtet, ohne dass eine Sicht abgelehnt wird. In den erweiterten Antworten werden die formulierten Anforderungen bestätigt und teilweise im Detail ergänzt. **[5.1 Editierbare Sichten aufs Datenmodell]**

Die formulierten Funktionen werden in den erweiterten Antworten einerseits bestätigt und andererseits im Detail ergänzt. Insbesondere auf die Unterstützung bei der Modellierung von Konsistenzbedingungen/Constraints wird grosser Wert gelegt. Zudem werden auch Änderungsverfolgung, Versionsmanagement, Changemanagement und die Wichtigkeit der Usability angesprochen. **[5.2 Auswahl von Objekten und Funktionen]**

Ob Sprachübersetzungen ausschliesslich in der speziellen Übersetzungstabelle geändert werden dürfen oder auch in den vier Hauptansichten wird mit  $\frac{2}{3}$  zu  $\frac{1}{3}$  beantwortet. **[5.3 Modellübersetzungen]**

Werte eines Katalogs sollen in Tabellenform erstellt, diskutiert und aktualisiert werden können. Bei der Frage ob zusätzlich auch eine editierbare XML-Sicht zur Verfügung stehen soll, sagt je ein Viertel JA bzw. Nein. Aber 50% möchten zusätzlich eine nicht editierbare XML Sicht auf den Katalog. Zudem wird darauf hingewiesen, dass Kataloge durchsucht und geprüft werden können sollten. **[5.4 Erstellen, editieren, übersetzen von Katalogen]**

Zu den Anforderungen zu Modellprüfung und Hilfefunktion gibt es Detailhinweise für die Unterstützung bei der Syntax und beim Erstellen und Prüfen von Constraints sowie zur Verbesserung der Sichtbarkeit bei Änderung. Zudem wurde gewünscht, dass ein Delta zu einer früheren Modellversion sichtbar gemacht werden kann. **[5.5 Modell prüfen, Hilfefunktion]**

Bei der Frage, ob die INTERLIS-Workbench auch eine Funktion zur Definition von Darstellungskatalogen beinhalten soll, sprechen sich  $\frac{2}{3}$  dafür und  $\frac{1}{3}$  dagegen aus. Eine solche Funktion ist einerseits wünschenswert, aber der Komplexität sind Grenzen gesetzt und die Interoperabilität wird stark hinterfragt. Wenn, dann eher in zweiter Priorität. **[5.6 Darstellungskatalog]**

Bei der Frage ob die INTERLIS-Workbench auch eine Funktion für die rechtliche und fachliche Prosabeschreibung zur Verfügung stellen soll, sprechen sich rund  $\frac{2}{3}$  dafür und  $\frac{1}{3}$  dagegen aus. Die Vorteile sehen die TeilnehmerInnen indem alle Elemente der Modelldokumentation an einem Ort erstellt und verwaltet werden können und bei der Aktualisierung der zentral gehaltenen Informationen. Die Ungewissheit bleibt bei der konkreten Ausgestaltung und bei komplexeren Inhalten (Bilder, Videos, etc.), sowie bei der Interoperabilität und dem effektiven Nutzen. Daher eher zweite Priorität. **[5.7 Modelldokumentation]**

Bei Import und Export werden automatisierte Aufbereitung der Daten in der Applikation, Drag&Drop und Nachvollziehbarkeit angesprochen. Auch eine Deklaration der unterstützten Sprachversionen wird empfohlen. Offene und Plug-In Architektur wird gewünscht, bzw. bestätigt. **[5.8 Modell und Modellbestandteile exportieren und 5.9 Modell oder Modellbestandteile importieren]**

Bei der Frage, welche Technologien und Kollaborationsmöglichkeiten als wichtig erachtet werden, um vernetztes, kollaboratives Arbeiten an den Modellen oder bei der Weiterentwicklung der Applikation zu gewährleisten, wird das parallele Arbeiten an Datenmodellen eher in Frage gestellt. Daher werden eher Standalone- und allenfalls zusätzlich eine Server-Lösungen vorgeschlagen. Weiter werden Git, OpenSource, Programmier-API erwähnt, bzw. bestätigt.

Trotz der Skepsis beim verteilten Modellieren wird eine Kollaboration für den Austausch unter ExpertInnen, bei internen und externen Vernehmlassungen, bei der Bewertung von Kommentaren, und allenfalls bei Variantenvergleichen und bei der Dokumentation von (geplanten) Änderungen gewünscht. **[5.11 Technologie und Kollaboration]**

Gemäss Umfrage sollte eine Funktionen zur Modell- bzw. Schematransformation mit Zuordnungstabellen durch die Softwarearchitektur in Form von Plugins möglich gemacht werden. **[5.12 Modelltransformationen]**

Unter weiteren Bemerkungen wird das Vorhaben, eine umfassende INTERLIS-Workbench zu bauen, erneut begrüsst. Es wird erwähnt, dass ein gutes Tool aber alleine noch keinen guten Modellierer ergibt. Es wird empfohlen, mehrere Sprachversionen zuzulassen. Es haben sich verschiedene Personen als Betatester zur Verfügung gestellt oder gewünscht sich direkt mit dem Projektteam austauschen zu können.

## 2.2 Bedürfnisnachweis für eine umfassende INTERLIS-Workbench

Die Auswertung der Umfrage zeigt deutlich, dass eine umfassende INTERLIS-Workbench begrüsst und eine rasche etappierte Umsetzung gewünscht wird - der Bedürfnisnachweis ist somit erbracht.

Diverse Hinweise und präzisierende Vorschläge aus der Umfrage wurden in dieses Konzept integriert. Somit stellt das Konzept eine breit abgestützte Grundlage dar.

Es ist zwingend, dass dieses Konzept auch in Französisch und allenfalls in Italienisch übersetzt wird

### 3 Ausgangslage

Der UML/INTERLIS-Editor ist zurzeit das Werkzeug, um INTERLIS-Modelle zu modellieren, die Syntax zu prüfen und als textuelle INTERLIS-Datei (\*.ili) auszugeben. Zudem kann ein UML-Diagramm oder ein Objektkatalog (als Hypertext) generiert werden. Die Bearbeitung der Werte von externen Katalogen ist mit dem UML/INTERLIS-Editor heute jedoch nicht möglich und die Übersetzung eines INTERLIS-Modells in eine andere natürliche Sprache ist umständlich.

Der UML/INTERLIS-Editor und dessen Handling entsprechen nicht mehr einer modernen User-Experience. Die Unterstützung für effizientes Modellieren fehlt, da beispielsweise eine performante Hilfefunktion fehlt und einfaches Drag&Drop nicht möglich ist. Es gibt auch keine Unterstützung bei der Definition von Constraints im Sinne eines intelligenten Wizards zum Beispiel und es kann nicht zwischen verschiedenen editierbaren Sichten des Modells umgeschaltet werden, wie beispielsweise zwischen hierarchischem Modellbaum, dem INTERLIS-Code, dem Objektkatalog und dem UML-Diagramm. Das Erstellen des UML-Diagramms ist umständlich und visuell nicht ansprechend. Zudem vergisst der Farbgenerator für die UML-Diagramme nach der Definition die RGB-Werte und die Funktion zur Änderung der Reihenfolge der Elemente im Modell ist etwas schwerfällig und wirkt nur auf den Output – also kein WYSIWYG. Es wird auch beklagt, dass das GUI-Verhalten zumindest auf MacOS fehlerhaft ist, wie beispielsweise, dass das Klicken des OK-Buttons die Veränderung häufig ignoriert. Es werden auch einige grundlegende Funktionen wie Undo/Redo vermisst.

Daher gibt es Modellierende, welche lieber direkt in einem einfachen Text-Editor die INTERLIS-Modelle tippen, um sie anschliessend mittels INTERLIS-Compiler zu prüfen und über den INTERLIS/UML-Editor das UML-Diagramm und allenfalls den Entwurf des Objektkatalogs zu erstellen.

Um die Modellierenden beim Erstellen von korrekten Modellen zu unterstützen, gibt es zudem diverse Hilfsmittel in Form von Dokumenten. So findet man unter <https://www.geo.admin.ch/de/geodatenmodelle> beispielsweise «Allgemeine Empfehlungen zur Methodik der Definition minimaler Geodatenmodelle», «Empfehlungen zum Umgang mit externen Katalogen», «Empfehlung für die Erstellung von externen Katalogen» und «Empfehlung Change Management für Minimale Geodatenmodelle». Des Weiteren finden man dort noch die Dokumentation der «Basismodule des Bundes für «minimale Geodatenmodelle» sowie die «Vorlage für die Modelldokumentation» und die «UML-Vorlage für die Modellierung im INTERLIS UML-Editor» - etc.

Nicht zu vergessen sind die unter <https://www.interlis.ch/dokumentation/interlis-2> zu findenden grundlegenden Dokumentationen wie das «INTERLIS Referenzhandbuch», das «INTERLIS Benutzerhandbuch» sowie das «INTERLIS Glossar» und die «Zusatzfunktionen».

#### 3.1 Erkenntnis

Der UML/INTERLIS-Editor ist für einige Modellierende ein solides Werkzeug, welches jedoch hinsichtlich User-Experience, Möglichkeiten, Unterstützung und Effizienz diverse Wünsche offen lässt. Abgesehen von einigen Modellierenden werden die Nutzenden nur ungenügend abgeholt. D.h. der heutige UML/INTERLIS-Editor eignet sich nicht sonderlich gut für die Kommunikation während des Modellierungsprozesses. Die Datenmodelle können in der Applikation weder übersichtlich und verständlich dargestellt, noch den Nutzenden anschaulich kommuniziert werden. INTERLIS als Modellierungssprache lässt sich Entscheidungsträgern und Fachleuten nicht überzeugend präsentieren, wodurch der Nutzen des modellbasierten Ansatzes zu wenig erkannt wird (dies ist insbesondere dann wichtig, wenn INTERLIS auch im internationalen Umfeld vermehrt promotet werden soll).

## 4 Kernziele

- a. Die neue INTERLIS-Workbench (Arbeitstitel) soll ein **vollumfängliches Werkzeug** werden, welches ein effizientes, flexibles, intuitives und unterstütztes **Modellieren, Visualisieren, Diskutieren und Dokumentieren von Datenmodellen** ermöglicht.
- b. Das Werkzeug soll eine **moderne und einfache User-Experience** garantieren. Die Handhabung soll einfach und verständlich sein – auch für Nicht-Informatiker. Die Einstiegsschwelle für die Benutzenden soll niedrig sein.
- c. Das neue Werkzeug muss **verschiedene editierbare Sichten auf das Datenmodell** ermöglichen, die **Modellübersetzung** in andere natürliche Sprachen vollumfänglich gewährleisten und gegebenenfalls das Erstellen, Editieren und Visualisieren der **Wertebereiche von externen Katalogen** effizient unterstützen.
- d. Die verschiedenen **Sichten** sollen auch **zur Diskussion der Modelle** mit verschiedenen Fachleuten dienen. Zudem **erhöhen** die verschiedenen Sichten die **Effizienz von INTERLIS- oder Datenmodellierungsschulungen**, wobei die Möglichkeit, zwischen Sichten des Modells zu wechseln, den Lerneffekt beschleunigt. Hierfür ist ein **performanter Hilfe- und Prüfmechanismus** zwingend. Auch **für Entscheidungsträger** wird der **Vorteil des modellbasierten Ansatzes** klarer, wenn sie ein Datenmodell nicht in INTERLIS-Code sehen, sondern in grafischer oder tabellarischer Form.
- e. **Mittels «IMPORTS» importierte Modelle** bzw. deren im Modell verwendeten Bestandteile sollen **über einen Schalter eingeblendet werden können**, um das komplette Modell sichtbar machen und verstehen zu können.
- f. Die **Hilfefunktionen** sollen, **allenfalls mit Hilfe von künstlicher Intelligenz**, also KI-basiert, ein intuitives Modellieren ermöglichen und **aufgrund von Stichworten** konkrete, qualitativ **hochwertige Vorschläge unterbreiten**, wie beispielweise beim Definieren von Constraints oder bei Erweiterungen/Vererbungen, etc. Zudem soll die Hilfefunktion die **Erkenntnisse aus den diversen Empfehlungen** (siehe Ausgangslage) aktiv während dem Modellieren **einbringen**. Verschiedene Funktionen sollen das Einhalten der Empfehlungen unterstützen.
- g. Zudem soll das Werkzeug das **nachhaltige Verwalten der Datenmodelle**, der Modelldokumentationen, der externen Kataloge und allenfalls der Darstellungsmodelle unterstützen und, soweit möglich, Einträge für Repositories auf Knopfdruck erstellen können.

### 4.1 Rahmenbedingungen

Die konkreten Anforderungen werden aufgrund der Erfahrungen aus der Community definiert, wobei ein initialer Input durch GeoStandards.ch eingebracht wird, um die Diskussion anzuregen. Die später für die Realisierung zu wählenden Technologien und Funktionalitäten müssen bestmöglich in die bestehende INTERLIS-Werkzeuglandschaft eingepasst werden, um einerseits redundante Entwicklungen zu vermeiden und höchstmögliche Synergien zu erzeugen. Weitere (technische) Rahmenbedingungen können über die zu erwartenden IT-(Ziel)Systemumgebungen / IT-Strategien (Bund, Kanton, etc.) wie auch schlussendlich die zur Verfügung stehenden (finanziellen) Mittel gesetzt werden.

### 4.2 Abgrenzung

Welche Anforderungen als MUSS- oder SOLL-Anforderungen in ein Pflichtenheft einfließen, wird zu einem späteren Zeitpunkt definiert. Ob der neue Editor auf dem heutigen UML/INTERLIS-Editor

aufbaut oder auf der Basis eines anderen, etablierten UML-/Entwicklungswerkzeugs entwickelt werden soll, wird zu einem späteren Zeitpunkt definiert. Zuvor müssen alle Anforderungen, insbesondere der Funktionsumfang der INTERLIS-Workbench geklärt sein. Dabei ist wichtig, WAS der neue INTERLIS-Modell-Editor können muss, aber noch nicht WIE die Umsetzung bis ins kleinste Detail zu erfolgen hat.

In diesem Konzept werden ausschliesslich die Datenmodelle und deren Wertebereiche betrachtet. Werkzeuge, welche die Geodaten als solches bearbeiten, prüfen oder konvertieren, sind nicht Teil dieser Überlegungen. Auch das Erstellen und Prüfen von Testdaten fällt nicht in den Bereich dieser Überlegungen.

Die umfassende Funktionalität für das Modellieren, Übersetzen und Verwalten der Modelle soll im neuen Werkzeug «INTERLIS-Workbench» zusammengefasst werden. Der INTERLIS-Compiler beispielsweise soll nach wie vor alleinstehend entwickelt und genutzt werden können, aber auch in die INTERLIS-Workbench integriert werden.

Es ist zu klären, ob eine Funktionalität für Modellkonvertierungen (Modellumbau) Bestandteil der INTERLIS-Workbench sein soll.

Weiter soll geklärt werden, ob ein Modul eines Grafik-, bzw. Symbol-Editors zur Verfügung stehen muss, um Darstellungsmodelle erstellen, verwalten und für Drittsysteme ausgeben zu können.



## 5 Anforderungen

In den folgenden Kapiteln werden ergänzend zu den Zielen die Funktionalität und allenfalls die Strukturen und Technologien beschrieben, welche in einem künftigen Werkzeug umgesetzt werden sollen. Grundsätzlich werden die INTERLIS-Versionen 2.3 und 2.4 unterstützt.

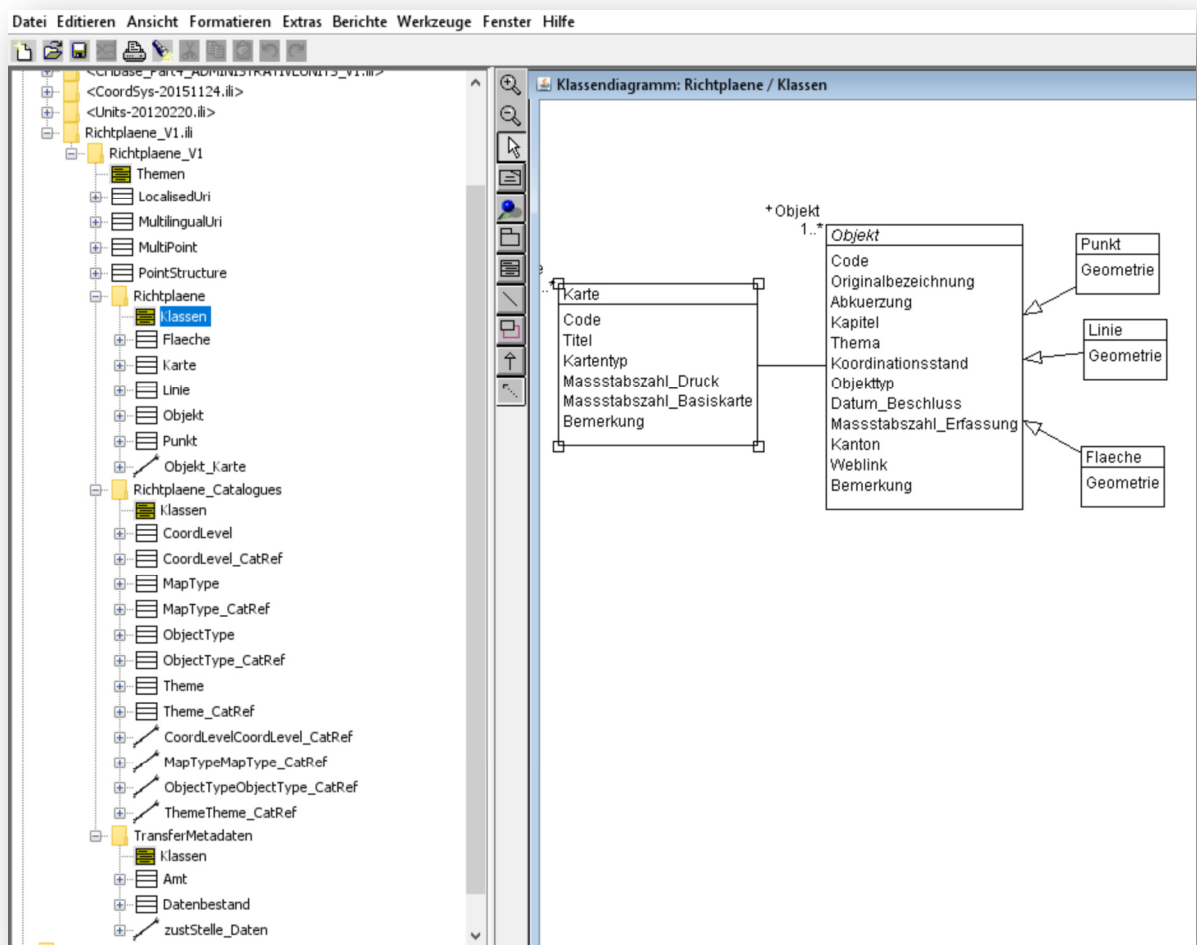
### 5.1 Editierbare Sichten aufs Datenmodell

Um alle Modellierenden abzuholen, aber auch die Kommunikation mit den Nutzenden zu gewährleisten, sollen verschiedene editierbare Sichten auf das Datenmodell zur Verfügung gestellt werden. Folgende Sichten sind vorgesehen:

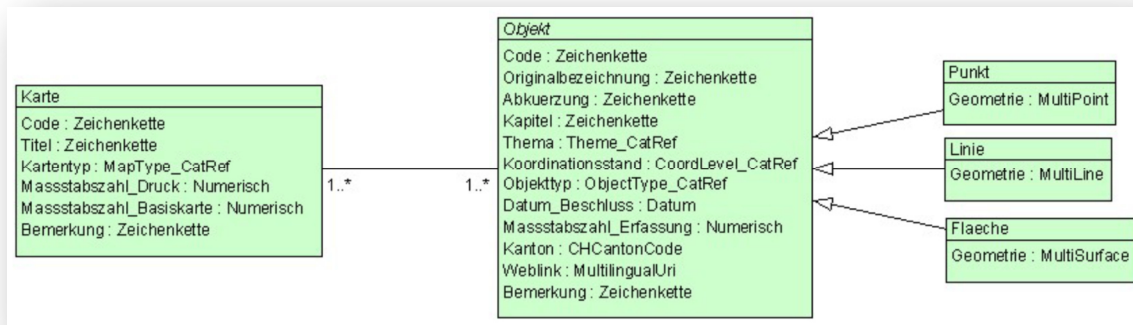
- Strukturbaum mit den spezifischen Editierfenstern (ähnlich dem heutigen UML/INTERLIS-Editor)
- Objektkatalog (Tabellenform; allenfalls zwei, drei Layouts)
- UML-Diagramm (mit automatisierter Anordnung der Diagrammelemente)
- Texteditor mit INTERLIS-Code, der die Funktionalität einer IDE (Integrated Development Environment) besitzt

Anschliessend werden Anschauungsbeispiele der Sichten auf der Grundlage des Datenmodells «Richtplanung» gezeigt.

Sicht auf Strukturbaum und UML-Diagramm, gemäss heutigem UML/INTERLIS-Editor:



Sicht des eingefärbten UML-Diagramms (aus der Modelldokumentation entnommen):



Sicht Objektkatalog (aus der Modelldokumentation entnommen):

**Klasse Objekt**  
 Die Klasse «Objekt» ist eine abstrakte Klasse. Sie enthält die Attribute der geometrischen Objekte, welche in den kantonalen Richtplänen vorkommen.

Name	Kardinalität	Typ	Beschreibung
Code	0..1	TEXT*50	Kantonaler Code des Objekts
Originalbezeichnung	1	TEXT*200	Originalbezeichnung des Objekts im kantonalen Richtplan
Abkuerzung	0..1	TEXT*25	Abkürzung des Objekts im kantonalen Richtplan
Kapitel	0..1	TEXT*200	Bezeichnung des Kapitels im kantonalen Richtplan
Thema	1	Thema_CatRef	Thema des Objekts. Jedem Objekt muss ein Thema zugeordnet werden, das «useable» ist (im Katalog <b>fett</b> gedruckt)
Koordinationsstand	1	CoordLevel_CatRef	Koordinationsstand des Objekts. Jedem Objekt muss ein Koordinationsstand zugeordnet werden, der «useable» ist (im Katalog <b>fett</b> gedruckt)
Objekttyp	1	ObjectType_CatRef	Typ des Objekts. Jedem Objekt muss ein Objekttyp aus dem Katalog zugeordnet werden.
Datum_Beschluss	1	XMLDate	Beschlussdatum der zuständigen kantonalen Behörde
Massstabszahl_Erfassung	0..1	100..1000000	Massstabszahl der Erfassung
Kanton	1	CHCantonCode	Kantonskurzzeichen
Weblink	0..1	MultilingualUri	Weblink (URI) für weitere Informationen in de, fr, it. Mit diesem Attribut kann ein Bezug zum Richtplantext hergestellt werden
Bemerkung	0..1	MTEXT*255	Bemerkung
Karte	1..*	Assoziation (Karte)	Karten, in denen die Objekte dargestellt werden

**Klasse Punkt**  
 Die Klasse «Punkt» spezialisiert die Klasse «Objekt» für Punktobjekte.

Name	Kardinalität	Typ	Beschreibung
Geometrie	1	MultiPoint	Punktgeometrie mit einem oder mehreren Punkten

**Klasse Linie**  
 Die Klasse «Linie» spezialisiert die Klasse «Objekt» für Linienobjekte.

Name	Kardinalität	Typ	Beschreibung
Geometrie	1	MultiLine	Liniengeometrie mit einer oder mehreren Linien

**Klasse Flaeche**  
 Die Klasse «Flaeche» spezialisiert die Klasse «Objekt» für Flächenobjekte.

Im Objektkatalog sollen auch Header-Informationen, Beziehungen, Konsistenzbedingungen/Constraints gezeigt werden, um die Modellbestandteile möglichst vollständig abbilden zu können. Denn der Objektkatalog und das UML-Diagramm sind die idealen Sichten aufs Modell, um das Modell mit

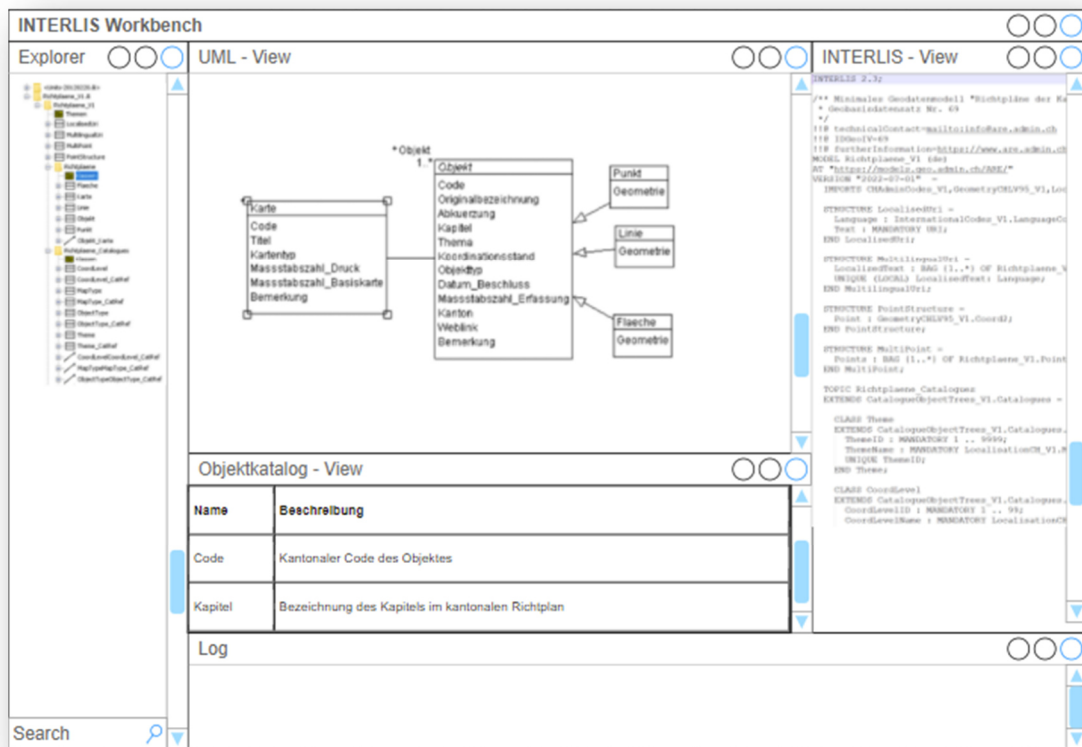
verschiedenen Fachleuten zu diskutieren. Bei einer online Anwendung oder Serverlösung könnten die Fachleute die Attribute und insbesondere deren Beschreibung dazu im Objektkatalog selber abfüllen.

### Sicht INTERLIS-Code «Richtplanung» (dargestellt im Notepad++):

```
Richtplaene_V1.ili
1 INTERLIS 2.3;
2
3 /** Minimales Geodatenmodell "Richtpläne der Kantone"
4  * Geobasisdatensatz Nr. 69
5  */
6 !!@ technicalContact=mailto:info@are.admin.ch
7 !!@ IDGeoIV=69
8 !!@ furtherInformation=https://www.are.admin.ch/mqm
9 MODEL Richtplaene_V1 (de)
10 AT "https://models.geo.admin.ch/ARE/"
11 VERSION "2022-07-01" =
12   IMPORTS CHAdminCodes_V1,GeometryCHLV95_V1,LocalisationCH_V1,CatalogueObjectTrees_V1,InternationalCodes_V1;
13
14   STRUCTURE LocalisedUri =
15     Language : InternationalCodes_V1.LanguageCode_ISO639_1;
16     Text : MANDATORY URI;
17   END LocalisedUri;
18
19   STRUCTURE MultilingualUri =
20     LocalisedText : BAG {1..*} OF Richtplaene_V1.LocalisedUri;
21     UNIQUE (LOCAL) LocalisedText: Language;
22   END MultilingualUri;
23
24   STRUCTURE PointStructure =
25     Point : GeometryCHLV95_V1.Coord2;
26   END PointStructure;
27
28   STRUCTURE MultiPoint =
29     Points : BAG {1..*} OF Richtplaene_V1.PointStructure;
30   END MultiPoint;
31
32   TOPIC Richtplaene_Catalogues
33   EXTENDS CatalogueObjectTrees_V1.Catalogues =
34
35     CLASS Theme
36     EXTENDS CatalogueObjectTrees_V1.Catalogues.Item =
37       ThemeID : MANDATORY 1 .. 9999;
38       ThemeName : MANDATORY LocalisationCH_V1.MultilingualText;
39       UNIQUE ThemeID;
40     END Theme;
41
42     CLASS CoordLevel
43     EXTENDS CatalogueObjectTrees_V1.Catalogues.Item =
44       CoordLevelID : MANDATORY 1 .. 99;
45       CoordLevelName : MANDATORY LocalisationCH_V1.MultilingualText;
46       UNIQUE CoordLevelID;
47     END CoordLevel;
48
49     CLASS MapType
```

Der Texteditor für INTERLIS-Code soll Funktionalitäten beinhalten, wie sie für IDE in der Softwareentwicklung zur Verfügung stehen, namentlich Syntax-Hervorhebung, automatische Vervollständigung von Schlüsselwörtern, Zugriff auf Modellelemente importierter Modelle und Sprungmarken zwischen Definitionen und der Anwendung von Modellelementen (z. B. Domains, Kataloge).

Nachfolgend ein Mock-Up, mit den auf Knopfdruck automatisch und kompakt angeordneten Sichten:



Diese vier Sichten aufs Modell sind nicht neu, sondern wohl bekannt. Neu ist, dass alle Sichten im Werkzeug dargestellt werden können und editierbar sind. Nicht jede Sicht wird alle Elemente bis ins letzte Detail darstellen können. Trotzdem sollen alle Sichten so vollständig, wie möglich sein, damit die Elemente ohne Weiteres in den anderen Sichten wiedererkannt werden. Für die Wiedererkennung soll auch die Reihenfolge der Elemente identisch sein.

Mit Hilfe von einem Auswahl-Menü sollen die Modellierenden eine oder mehrere Sichten auswählen können, welche aktiv angezeigt werden. Das Aktualisieren erfolgt jedoch unabhängig davon. Nebst der kompakten Darstellung sollen die Fenster der einzelnen Sichten frei positioniert und in der Gösse angepasst werden können, was insbesondere bei mehreren oder grossen Bildschirmen das Arbeiten komfortabler macht.

#### Modell - Sichten

- Strukturbau
- UML-Diagramm
- Objektkatalog
- INTERLIS-Code

Wenn in einer Sicht editiert wurde, sollen die anderen Sichten direkt aktualisiert werden. Zudem sollen auch jeweils die Syntax und weitere Bedingungen geprüft werden (s. auch Kap. Modell prüfen). Das Aktualisieren der weiteren Sichten und das Prüfen des Modells «on the fly» oder «on demand» sollen die Modellierenden allenfalls über einen Schalter einstellen können. Dort wo aktuell editiert wird, sollen Funktionen wie «undo», Rechtschreibung, Kommentarfunktion, etc. jederzeit zur Verfügung stehen. Alle Sichten basieren auf dem vollumfänglichen Inhalt der internen Verwaltung des neuen Tools. So wird beim Bearbeiten der Inhalt der internen Verwaltung laufend aktualisiert und die übrigen Sichten werden anschliessend daraus aktualisiert und das Modell geprüft (kein Konfliktmanagement nötig).

Die Bezeichnungen der Modellelemente sollen in allen Sichten identisch sein und wenn immer möglich der INTERLIS-Spezifikation entsprechen (z.B. heute in einer Sicht «TEXT\*», in einer anderen Sicht

- «Zeichenkette»). Und die Modellierenden müssen die Reihenfolge der Elemente, wie Attribute, Klassen, etc., beispielsweise durch Drag&Drop im Strukturbaum steuern können.

In der heutigen Version des UML/INTERLIS-Editor werden erläuternde Texte in der ili-Datei in der Form (/\*\* <Kommentar> \*/) umgesetzt, da dies der ilidoc-Konvention entspricht und solche Texte auch in den Objektkatalog integriert werden. In der neuen Applikation ist zu definieren, wie mit der Syntax für Zeilenkommentare (!! <Kommentar>) für erläuternde Texte umgegangen wird. Es soll gewährleistet werden, dass die INTERLIS-Workbench alle durch INTERLIS zugelassenen Syntax-Varianten für Kommentare unterstützt.

Zur Vollständigkeit sollen mittels «IMPORTS» importierte Modelle bzw. deren im Modell verwendeten Bestandteile über einen Schalter eingeblendet werden können, um das komplette Modell sichtbar machen und verstehen zu können. Umgekehrt soll es einen weiteren Schalter geben, welcher eine vereinfachte Modellansicht, fokussiert auf Basis-Klassen ermöglicht. Bei komplexeren Modellen könnte beispielsweise die Sicht aufs UML-Diagramm auf die Kernstrukturen reduziert werden.

Jede der Sichten soll auch separat in den jeweils gängigen Formaten exportiert werden können (abgesehen vom Strukturbaum). Es ist jedoch wichtig, dass die Verwaltung des Modells mit der INTERLIS-Workbench geschieht, da nebst dem reinen INTERLIS-Code weitere Informationen verwaltet werden, welche nicht verloren gehen dürfen.

## 5.2 Auswahl von Objekten und Funktionen

Es sollen verschiedene Objekte und Funktionen in kontextsensitiven Menüs zur Verfügung gestellt werden (ähnlich den Funktionen in Excel), um ein effizientes und korrektes Modellieren zu unterstützen. So soll beispielsweise ein Objekt mit einer bestimmten Geometrie oder ein Objekt mit verschiedenen Geometrien ausgewählt werden können. Auch seltenere und komplexere Datenstrukturen und Beziehungen, wie beispielweise bei VIEWS, Vererbungen, etc. sollen aufgrund von Funktionen generiert werden können. Die Modellierenden sollen mit Hilfe der ausgewählten Funktion durch die korrekte Erstellung geführt werden – insbesondere dann, wenn es von den Modellierenden verschiedene weitere Angaben braucht. Letztlich soll jeweils die ganze benötigte Struktur erstellt und deren korrekte Einbindung ins Modell gewährleistet werden.

Auch bei den Konsistenzbedingungen/Constraints sollen die Modellierenden stark unterstützt werden. Wünschenswert wäre eine Funktion in Form eines KI-basierten Constraint-Editors (Wizard), welcher aus einer kompakten und korrekten Prosabeschreibung einer Bedingung einen korrekten Constraint formuliert. Zur Kontrolle soll die KI den Constraint wieder als Bedingung in Prosa beschreiben. Dies ist notwendig, da ein guter Constraint-Editor die kompakteste Variante wählt und sich nicht ausschliesslich auf die «Straight-forward-Prosa-Beschreibung» der Ausgangslage stützt. Die Constraints sollen auf Knopfdruck separat gegen die Bedingung geprüft werden können. Eine solche Prüfung soll immer auch wieder die Bedingung in Prosa beschreiben. Aber auch ohne KI kann eine Unterstützung bereits hilfreich sein, welche als Input «if/then/else» sowie «and/or» korrekt umsetzt. Da Constraints schwer zu interpretieren sind (führt immer wieder zu Missverständnissen), wäre ein Pop-up-Fenster zum Prüfen und Spielen hilfreich, wann ein Constraint «TRUE» und wann «FALSE» zurückgibt.

Es wäre in der heutigen Zeit auch vorstellbar, dass eine Funktion auf Grund einer Prosabeschreibung mit den fachlichen Anforderungen der Vollzugsaufgabe mittels KI automatisch einen ersten Datenmodellentwurf generiert. KI kann heute schon ein INTERLIS-Referenzhandbuch, bzw. dessen codierte Definitionen interpretieren und anwenden. Anschliessend könnte der Entwurf von den Modellierenden geprüft, korrigiert und allenfalls ergänzt werden. Eine solche Funktion wäre nicht nur sehr

effizient für erfahrene Modellierende, sondern würde die Einstiegsschwelle in die Welt des Modellierens, wie in den Zielen gefordert vereinfachen.

Es sollen auch Funktionen zur Versionierung der Modelle, u.a. gemäss Change-Management, und Funktionen zur Änderungsverfolgung bereitgestellt werden. Das Visualisieren von Änderungen zwischen zwei Modellversionen soll ermöglicht werden.

Im Sinne des verteilten Arbeitens soll eine Kommentarfunktion die Kommunikation unter den am Modellierungsprozess beteiligten Personen verbessern.

Funktionen wie «Modell übersetzen» oder «Katalog erstellen» werden in den folgenden Kapiteln ausführlicher diskutiert.

## 5.3 Modellübersetzungen

### 5.3.1 Sicht auf die Übersetzungstabellen

Modellübersetzungen von einer natürlichen Ausgangssprache in eine oder mehrere andere (TRANSLATION OF), müssen direkt im Modell-Editor erstellt und verwaltet werden können, inkl. VIEWS. Vermutlich wird sich die Sicht des tabellenförmigen Objektkatalogs mit reduzierten Spalten am besten für die Übersetzungen eignen, sowohl fürs Editieren, als auch für die Diskussion mit Fachleuten aus anderen Bereichen und anderen Sprachregionen. Ob sich die Sicht des Objektkatalogs definitiv eignet oder ob es eine separate tabellarische Sicht braucht, muss noch abschliessend geklärt werden.

Übersetzt werden nebst der Modellstruktur mit den Attribut- und Klassenbezeichnungen, etc. auch die Erläuterungen zu den Strukturbestandteilen und allenfalls die Wertebereiche. Auch die Header-Sektion, allenfalls die Beziehungen und Konsistenzbedingungen/Constraints müssen übersetzt werden können. Hierfür soll diese Information auch in der Sicht für die Übersetzungen eingeblendet werden.

Beispiel mit Deutsch und Französisch (Auszug aus den Objektkatalogen der Modelldokumentation):

Name	Beschreibung	Nom	Description
Code	Kantonaler Code des Objekts	Code	Code cantonal de l'objet
Originalbezeichnung	Originalbezeichnung des Objekts im kantonalen Richtplan	DesignationOriginale	Désignation originale de l'objet dans le plan directeur cantonal
Abkürzung	Abkürzung des Objekts im kantonalen Richtplan	Abreviation	Abréviation de l'objet dans le plan directeur cantonal
Kapitel	Bezeichnung des Kapitels im kantonalen Richtplan	Chapitre	Désignation du chapitre dans le plan directeur cantonal
Thema	Thema des Objekts. Jedem Objekt muss ein Thema zugeordnet werden, das «useable» ist (im Katalog <b>fett</b> gedruckt)	Theme	Thème de l'objet. Un thème « useable » doit être attribué à chaque objet (figure <b>en caractères gras</b> dans le catalogue)
Koordinationsstand	Koordinationsstand des Objekts. Jedem Objekt muss ein Koordinationsstand zugeordnet werden, der «useable» ist (im Katalog <b>fett</b> gedruckt)	EtatDeCoordination	Etat de coordination de l'objet. Un état de coordination « useable » doit être attribué à chaque objet (figure <b>en caractères gras</b> dans le catalogue)
Objekttyp	Typ des Objekts. Jedem Objekt muss ein Objekttyp aus dem Katalog zugeordnet werden.	TypeDObjet	Type d'objet. Un type d'objet du catalogue doit être attribué à chaque objet.
Datum_Beschluss	Beschlussdatum der zuständigen kantonalen Behörde	Date_Decision	Date de la décision prise par l'autorité cantonale compétente
Massstabszahl_Erfassung	Massstabszahl der Erfassung	Echelle_Saisie	Echelle de la saisie
Kanton	Kantonskurzzeichen	Canton	Abréviation du canton
Weblink	Weblink (URI) für weitere Informationen in de, fr, it. Mit diesem Attribut kann ein Bezug zum Richtplantext hergestellt werden	LienWeb	Lien (URI) pour d'autres informations en de, fr, it. Cet attribut permet de créer un lien avec le texte du plan directeur
Bemerkung	Bemerkung	Remarque	Remarque
Karte	Karten, in denen die Objekte dargestellt werden	Carte	Cartes dans lesquelles les objets sont représentés

Die Modellierenden sollen entscheiden können, wie viele zusätzliche Sprachen, nebst der primären Sprache, in der Sicht für die Übersetzungen aktiv angezeigt werden sollen (mindestens die primäre Sprache und jeweils eine der zu ergänzenden Sprachen). Die Auswahl soll auch hier über ein Auswahlmenu erfolgen. Die Übersetzung erfolgt automatisiert, indem vorzugsweise Text aus der primären Sprache KI-basiert in die zu ergänzende Sprache übersetzt und als Vorschlag in die Tabelle eingefügt wird. Der Vorschlag soll anschliessend manuell verbessert werden können

### **5.3.2 Sicht auf das gesamte Modell in einer bestimmten Sprache**

Nebst der Übersetzungsfunktion in der tabellarischen Sicht soll jeweils ein gesamtes Modell mit allen Sichten in einer bestimmten Sprache dargestellt werden können. Ob die Editierfunktion bei der Auswahl eines übersetzten Modells eingeschränkt (d.h. nur Attributnamen, Aufzählungen und deren Beschreibungen sind editierbar) oder ganz gesperrt sein soll, muss noch geklärt werden, da grundsätzlich das Modell in der primären Sprache editiert wird und die Übersetzungen danach erfolgen. Ein eingeschränktes Editieren kann den Komfort erhöhen, aber Konflikte zwischen den verschiedenen Sprachen begünstigen. Bei einer gesperrten Editierfunktion müssten alle Änderungen mittels der Übersetzungsfunktion gemäss Kap 5.3.1 erfolgen, welche nur den Weg von der primären Sprache zu der übersetzten Sprache zulässt.

Zellen, welche in der primären Sprache geändert wurden, sollen in den Übersetzungen (Objektkatalog) in jedem Fall gekennzeichnet werden, um zu gewährleisten, dass die Änderungen in den Übersetzungen nicht vergessen gehen.

## **5.4 Erstellen, editieren, übersetzen von Katalogen**

Kataloge sollen vollständig in der neuen INTERLIS-Workbench erstellt und verwaltet werden können. Es soll wählbare Funktionen geben, welche das Erstellen von verschiedenen Katalogausprägungen ermöglichen. Die entsprechenden Kataloge sollen immer gleich aufgebaut sein. Ausgangspunkt ist immer die zu erstellende/editierende Werteliste. So wählen die Modellierenden die Funktion «Katalog erstellen» und definieren zu Beginn die Anzahl Attribute, bzw. Spalten der Wertetabelle. In Katalogen kann ein Wert mehrere Ausprägungen enthalten, wie ein Code, eine Kurzbezeichnung, eine Alias und allenfalls eine Beschreibung. Dies muss bei den Übersetzungen, beim Aufbau und der Erstellung der Katalogtabelle berücksichtigt werden. Es ist zu prüfen, ob diese Funktionen das entsprechende Modell aus den Basismodulen ersetzen kann, da die Funktion die Struktur direkt erstellt.

Der Katalog bzw. die Werte des Katalogs sollen grundsätzlich in Tabellenform bearbeitet und diskutiert werden können. Werte aus bestehenden externen Dokumenten, wie beispielsweise Codelisten mit Erläuterungen, sollen mittels Copy/Paste oder durch einen Import in die Tabelle der INTERLIS-Workbench übernommen werden können. Die heute schon verwendete Auswahlmöglichkeit von Parent und/oder Child-Werten soll weiterhin möglich und einfach editierbar sein. Die Tabellenform für die Sicht auf die Werte des Katalogs eignet sich auch hier wieder für die Übersetzungen. Auch bei Katalogen sollen die zusätzlichen Sprachen mittels Auswahlmenu angezeigt und editiert werden können. Grundsätzlich verhält sich die Übersetzung der Kataloge ähnlich jener der Modellübersetzungen (Kap. 5.3.1). Die Kataloge sollen auch durchsucht werden können und es braucht Prüfmechanismen, damit man nur gültige Kataloge erstellt (OID, ...)

Eine Mehrheit aus der Umfrage wünscht, dass der Katalog auch in der Ausprägung XML angezeigt werden kann. Auf eine editierbare XML Sicht soll jedoch verzichtet werden, um nicht Konflikte aus fehlerhaften XML-Einträgen lösen zu müssen. Auch bei der XML-Sicht sollen nur die zugelassen

Ausprägungen/Strukturen ermöglicht werden. Als Export ist XML, bzw. XTF nebst gängigen Tabellenformaten zwingend. Es wird jeder Katalog separat verwaltet. Für den Export soll geprüft werden, ob alle Kataloge in eine Datei zu schreiben, zugelassen werden soll. Allenfalls könnte der Entscheid auch dem User mittels einer entsprechenden Funktion überlassen werden.

Hinweise: Bei einem Katalog besteht immer auch das Potential, dass er durch ein anderes Modell genutzt werden kann; ein ganzes Set von Katalogen in einer Datei beinhaltet dieses Potential eher weniger, da das Erkennen des spezifischen Katalogs schwieriger wird. Es soll gewährleistet sein, dass ein Katalog durch ein oder mehrere Modelle benutzt werden kann und nicht umgekehrt. So könnte ein Katalog mit Kantonsnamen und -kürzel oder ein Katalog mit Gemeindenamen, inklusive Übersetzungen, in vielen Modellen verwendet werden. Ob es dann noch ein entsprechendes Modell aus den Basismodulen braucht, wäre zu klären.

## 5.5 Modell prüfen, Hilfefunktion

Wie schon in den vorherigen Kapiteln beschrieben, soll das Modell «on the fly» oder «on demand» geprüft werden können.

Die Syntaxprüfung soll analog der Rechtschreibung in Word aufgebaut sein, indem direkt Korrekturvorschläge eingeblendet werden. Die Prüfung soll nebst der reinen Syntax von INTERLIS **auch Vorgaben aus Empfehlungen prüfen** (s. Kap. Ausgangslage) und dabei konkrete Hinweise für die Korrektur/Ergänzung vorschlagen. Vermutlich gelten nicht alle Empfehlungen für alle Modelle. So können beispielweise minimale Geodatenmodelle zusätzliche Anforderungen beinhalten, welche für andere Modelle nicht zwingend gelten. Es ist zu klären, ob Korrekturvorschläge aus allen Empfehlungen erfolgen und von den Modellierenden bewertet und akzeptiert oder verworfen werden, oder ob die Empfehlungen vorgängig für ein bestimmtes Modell ausgewählt werden müssen, so dass die Prüfung nur aufgrund dieser Auswahl Vorschläge unterbreitet. Der User soll aber entscheiden können, ob er nur gegenüber dem Compiler prüfen oder auch die Empfehlungen einbeziehen möchte.

Bei der Prüfung, der Hilfefunktion und bei der Auswahl von Funktionen werden von der Applikation konkrete Vorschläge, mindestens aber Hinweise erwartet. Daher ist es zwingend, dass diese Funktionen und deren KI identische Vorschläge unterbreiten. Es wäre äusserst schlecht, wenn sich beispielweise die Vorschläge oder Hinweise bei ausgewählten Funktion und bei Prüffunktionen oder bei der Hilfefunktion widersprechen würden.

Bei neueren Modellversionen wäre es sehr hilfreich, wenn mittels einer Funktion das Delta zur älteren Version sichtbar gemacht werden könnte.

## 5.6 Darstellungskatalog

Wenn Geodaten zugänglich gemacht werden müssen, beispielweise mittels eines Darstellungsdiensts, stellt sich wie bei minimalen Geodatenmodellen die Frage nach einem einheitlichen Darstellungsmodell. Auch wenn Geodaten nur aufgrund von Testdaten eines neuen Geodatenmodells betrachtet werden sollen, taucht sofort auch die Frage nach der Darstellung auf.






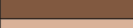







Fachinformationsgemeinschaften, welche neue Geodatenmodell diskutieren und Testdaten erstellen müssen, werden in der Regel auch die Darstellung diskutieren. Dann wird jemand in seinem GI-System die Testdaten erstellen und einen ersten Entwurf des Darstellungsmodells erarbeiten.

Es wäre daher wünschenswert, wenn mit einem neuem Modellierungstool auch gleich das einheitliche Darstellungsmodell definiert und in die gängigsten Formate exportieren werden könnten. Hierfür



müsste die neue INTERLIS-Workbench einen Darstellungs- bzw. Symboleditor beinhalten. Eine denkbare Implementierung wäre, wenn man im UML-Klassendiagramm bei den einzelnen Klassen Darstellungsinformationen hinterlegen könnte. Also Regeln zur Stilisierung (Query) als auch grafische Stildefinition und deren Implementierung als SLD-Codeschnipsel sowie als Stildateien von GI-Systemen (z. B. QGIS QML Style File). Als grafischer Output müsste mindestens eine Symbollegende erzeugt werden, welche in tabellarischer Form Symbole, RGB-Werte (allenfalls auch weitere Farbdefinitionen), den Bezug zu den Geoobjekten und Erläuterungen zum Symbol, etc. zeigt.

Beispiel aus der Modelldokumentation «Richtplanung»:

Code Thema	Thema 1	Thema 2	Farbcode	R	G	B	Farbmuster
101	Siedlung	Siedlungsgebiet	C13	255	242	0	
102	Siedlung	Bauzonen	C15	255	166	0	
103	Siedlung	Zentrenstruktur	C04	89	51	25	
104	Siedlung	Entwicklungsschwerpunkte	C18	230	0	0	
105	Siedlung	Verkehr-intensive Einrichtungen	C16	255	77	0	
106	Siedlung	Ortsbildschutz	C03	128	89	64	
107	Siedlung	Kleinsiedlungen (Art. 33 RPV)	C01	217	178	153	
108	Siedlung	Gebiete mit traditioneller Streubauweise...	C12	255	255	140	
109	Siedlung	Landschaftsprägende Bauten	C14	242	204	115	
110	Siedlung	Kulturdenkmäler von kantonaler Bedeutung	C02	179	128	102	
111	Siedlung	Halteplätze für Jenische, Sinti und Roma	C17	245	164	159	
119	Siedlung	Weitere Themen	C18a	255	63	63	
201	Natur und Landschaft	Landschaftstypen und -struktur	C05	0	149	67	

Das Darstellungsmodell, welches die Symbole konkret mit den Objekten oder Objektbestandteilen verbindet, soll als Ganzes mit dem Datenmodell zusammen verwaltet werden können, da eine Datenmodelländerungen u.U. auch eine Anpassung des Darstellungsmodells nach sich zieht. Die Darstellungsdefinition soll in gängigen Formaten exportiert werden können. Zudem soll die Barrierefreiheit gewährleistet werden, indem geprüft werden kann, ob die Farben auch bei Farbenfehlsichtigkeit unterschieden werden können.

## 5.7 Modelldokumentation

Die Geodatenmodelle und die Kataloge werden heute in INTERLIS-Model-Repositories, wie beispielsweise «<https://models.geo.admin.ch>» verwaltet. Bei den Modelldokumentationen gibt es heute verschiedene Lösungen. Teilweise findet man die Dokumente (PDF) auch in den Model-Repositories, oft werden sie jedoch an völlig unterschiedlichen Orten verwaltet und zugänglich gemacht.

Wenn in der neuen INTERLIS-Workbench bereits die wichtigen Bestandteile einer Modelldokumentation wie das UML-Diagramm, der Objektkatalog, der INTERLIS-Code und allenfalls das Darstellungsmodell erstellt und verwaltet werden können, stellt sich unweigerlich die Frage, weshalb die INTERLIS-Workbench nicht auch noch eine Möglichkeit bietet, den Prosatext mit Bildern, Grafiken und Videos zu erstellen, um alle Bestandteile einer Modelldokumentation mit dem gleichen Tool bearbeiten und verwalten zu können.

Die Modelldokumentation hätte so folgende Inhalte:

1. Prosatext mit rechtlicher Grundlage und fachlichen Anforderungen
2. UML-Diagramm als grafische Darstellung des Modells
3. Objektkatalog als tabellarische Ansicht des Modells
4. Gegebenenfalls Kataloge

5. Darstellungsmodell in Form einer erweiterten Symbollegende und allenfalls Zuordnungstabelle → Symbol – Geoobjekt-(Bestandteil)
6. INTERLIS-Code

*(weitere Bestandteile sind allenfalls zu diskutieren)*

Aus Gründen der Nachverfolgbarkeit und der kollaborativen Zusammenarbeit bietet es sich an, für die zukünftige Modelldokumentation eine Auszeichnungssprache wie Markdown oder AsciiDoc zu verwenden. Diese archaisch anmutende Erfassung von reinem ASCII-Text hat den Vorteil, dass sich Änderungen gut dokumentieren lassen (Versionierung) und es besteht die Möglichkeit, automatisiert abgeleitete Dokumentenformate wie PDF, Word oder ODF sowie auch andere Auszeichnungssprachen wie HTML zu erzeugen. Eine zusätzliche WYSIWYG-Ansicht hilft bei der Anwendung der Auszeichnungs-Optionen im Text-Editor.

## **5.8 Modell und Modellbestandteile exportieren**

Alle sechs Modellbestandteile müssen in geeigneten Formaten exportiert werden können, um diese Definitionen beispielsweise in anderen Applikationen weiterverarbeiten zu können. Auch die Modelldokumentation als Ganzes müsste als Dokument exportiert werden können.

Im Repository müssten vermutlich alle sechs oder sieben Bestandteile des Modells explizit vorliegen und die aktuelle Datei der INTERLIS-Workbench würde allenfalls auch dort abgelegt. Um die Zuordnung des Modells zum Repository definieren zu können, soll in der Applikation mittels einer Funktion der komplette, hierarchische Repository-Baum angezeigt werden.

Die Einträge fürs Modell-Repository sollen automatisiert mit der INTERLIS-Workbench erstellt und aktualisiert werden. Hierbei sind wie bereits beim Modellieren die Empfehlungen des Change-Management und weitere Empfehlungen oder Umsetzungshilfen zu berücksichtigen und einzuhalten.

## **5.9 Modell oder Modellbestandteile importieren**

Entsprechend der Funktion «Exportieren», sollen auch Modell und Modellbestandteile importiert werden können. So sollen beispielsweise bestehende Modelle, die ausschliesslich in Form der ili-Datei vorliegen, importiert werden können, wobei die anderen Sichten aufs Modell automatisch erstellt werden. Wie schon früher erwähnt, sollen externe Wertetabellen importiert und automatisch in einen Katalog integriert werden können. Die Modelle und Modellbestandteile sollen auch im Drag&Drop Verfahren importiert werden können.

## **5.10 Unterstützung von Validierungsmodellen**

Validierungsmodelle dienen der Auslagerung von Constraints in separate INTERLIS-Modelle, so dass die Daten-Validierung getrennt vom Datenmodell betrachtet und erarbeitet werden kann. Die INTERLIS-Workbench soll Unterstützung bei der Erarbeitung und beim Testen von Validierungsmodellen bieten, was bedeuten würde, dass Validierungsfunktionen nicht nur für Modelle (.ili) sondern auch für Daten (.xtf) zur Verfügung stehen müssten.

## 5.11 Technologie und Kollaboration

Die neue INTERLIS-Workbench könnte als App/Service und somit Online-Anwendungen entwickelt werden oder aber als Stand-Alone Software, welche lokal auf den Computern installiert wird und die Daten auch dort verwaltet werden.

Der Workflow und die Kompatibilität mit anderen Tools sollen gewährleistet sein. Hierfür soll es auch ein performantes API geben, um beispielsweise Schnittstellen für die automatisierte Weiterverarbeitung zu ermöglichen. Um benutzerdefinierte Erweiterungen und Anpassungen zu ermöglichen, drängt sich eine Plugin-Architektur auf. Eine solche Architektur in Ergänzung zum API, unterstützt ein Weiterentwickeln dieser Plattform. So könnte beispielsweise aus einem Modell im Repository mittels einer zusätzlich entwickelten Plugin-Funktion, direkt in ein konkretes DB-Schema erzeugt werden (z.B. via ili2pg, ili2gpkg etc.). Ein anderes Beispiel wäre Metadaten zu den MGDM erfassen und publizieren zu können im Sinne einer Interoperabilität zum Geometadaten-Katalog, etc.

Die «interne Datenverwaltung» der INTERLIS-Workbench sollte ebenfalls als INTERLIS-Modell beschrieben sein. Die internen Daten könnten dann in einer GeoPackage-Datei persistiert oder in einem .xtf in einem Repo abgelegt werden. Das würde zudem die Versionierung mittels git vereinfachen.

Die Performance soll auch mit grossen Modellen ein effizientes Arbeiten ermöglichen.

Die Umfrage hat gezeigt, dass eher ein Standalone-Werkzeug mit API bevorzugt wird (allenfalls zusätzlich als Serverlösung).

Um die Kollaboration zu unterstützen, soll die INTERLIS-Workbench auf einem bewährten verteilten Versionsverwaltungssystem wie git aufbauen. Damit liessen sich alle erzeugten Dateien (Modelldokumentation, Datenmodelle und deren Ausprägung in UML, Objektkataloge, Kataloge und Darstellungsmodelle) einheitlich verwalten und dokumentieren. Durch die Bereitstellung der Dateien in einem Software Repository bei einem Online-Dienst wie GitHub oder GitLab liessen sich auch deren Kollaborationswerkzeuge wie Issue-Tracker oder Diskussionsforen nutzen.

## 5.12 Modelltransformationen

Es stellt sich grundlegend die Frage, ob die INTERLIS-Workbench auch Funktionen zur Modell- bzw. zur Schematransformation mit Zuordnungstabellen beinhalten soll oder ob Schematransformationen in separaten Werkzeugen erfolgen sollen, wie beispielsweise mit FME.

Gemäss Umfrage sollte eine Funktionen zur Modell- bzw. zur Schematransformation mit Zuordnungstabellen durch die Softwarearchitektur in Form von Plugins möglich gemacht werden.

Die Geodatenkonvertierung als solches ist jedoch gemäss Kap. Abgrenzung nicht Teil dieser Überlegungen.

## 6 Glossar

Dieses Glossar bietet eine Übersicht über die wichtigsten Begriffe, die im Zusammenhang mit einer INTERLIS-Workbench relevant sind. Es soll als schnelle Referenz dienen, um das Verständnis der Kernkonzepte zu erleichtern.

### **Dateninfrastruktur**

Die technischen und organisatorischen Rahmenbedingungen, die den Austausch, die Speicherung und die Verarbeitung von Geodaten ermöglichen. Eine gut entwickelte INTERLIS-Workbench trägt zur effizienten Modellierung solcher Infrastrukturen bei.

### **Datenmodellierung**

Der Prozess der Erstellung eines Datenmodells, das die Struktur, Beziehungen und Regeln von Daten innerhalb eines Systems beschreibt. In einer INTERLIS-Workbench wird dies durch die Erstellung von UML-Diagrammen und weiteren Sichten aufs Modell unterstützt.

### **Geodaten**

Räumliche Daten, die Informationen über geografische Objekte und deren Standorte enthalten. Geodaten sind der Hauptfokus von INTERLIS und werden in Geoinformationssystemen (GIS) verwendet.

### **INTERLIS**

Ein standardisiertes Austauschformat und eine Modellierungssprache für Geodaten in der Schweiz. Es wird verwendet, um Geodatenmodelle zu beschreiben und den Austausch von Geoinformationen zwischen verschiedenen Systemen zu ermöglichen.

### **Multi-User-Support**

Eine Funktion, die es mehreren Benutzern ermöglicht, gleichzeitig an einem Modell zu arbeiten. Dies ist besonders wichtig in kollaborativen Umgebungen, in denen Teams zusammen an Geodatenmodellen arbeiten.

### **Plugin-Architektur**

Ein flexibles System, das es ermöglicht, zusätzliche Funktionen oder Erweiterungen in eine Software zu integrieren. Eine INTERLIS-Workbench mit Plugin-Architektur erlaubt es Benutzern, das Tool um spezifische Funktionen zu erweitern.

### **Schnittstelle (API)**

Eine Programmierschnittstelle, die es verschiedenen Softwareanwendungen ermöglicht, miteinander zu kommunizieren. In einer INTERLIS-Workbench können APIs genutzt werden, um Datenmodelle nahtlos in andere Systeme zu integrieren.

### **UML (Unified Modeling Language)**

Eine standardisierte Modellierungssprache, die zur visuellen Darstellung und Spezifikation von Softwaresystemen verwendet wird. Im Kontext von INTERLIS dient UML dazu, komplexe Datenstrukturen und deren Beziehungen in Form von Diagrammen darzustellen.

### **UML-Diagramm**

Ein grafisches Diagramm, das in UML verwendet wird, um die Struktur und das Verhalten eines Systems zu modellieren. Beispiele sind Klassendiagramme, Anwendungsfalldiagramme und Sequenzdiagramme.

### **Validierung**

Der Prozess der Überprüfung eines Datenmodells oder einer Datenstruktur auf Fehler und Inkonsistenzen. Eine INTERLIS-Workbench sollte Werkzeuge zur automatischen Validierung von Modellen bieten, um sicherzustellen, dass sie den Standards entsprechen.

## **Versionierung**

Das Verfolgen und Speichern von Änderungen an einem Modell über verschiedene Versionen hinweg. Dies ermöglicht es Benutzern, frühere Versionen wiederherzustellen und den Verlauf der Modellentwicklung nachzuvollziehen.