FGDM4GS

"Flat Geodata Models for Geoservices".

Konzeptmodellbasierter Ansatz für die Implementierung von Geodiensten für MGDM (Attribute und Symbologie)

Autoren: Maxime Collombin, Olivier Ertz, Jens Ingensand

Datum: Okt. 2024

Inhaltsverzeichnis

1. Hintergrund und Problematik

Als Geobasisdaten im Sinne von Art. 3[1] des Bundesgesetzes über Geoinformation (GeoIG) gelten Geodaten, die sich auf einen Erlass des Bundes, eines Kantons oder einer Gemeinde stützen. Zudem schreibt gemäss Art. 9[2] der Geoinformationsverordnung (GeoIV) die zuständige Fachstelle des Bundes ein minimales Geodatenmodell (MGDM) vor. Darin legt er die Struktur und den Grad der inhaltlichen Spezifikation fest. Und gemäss dem Dokument des Umsetzungsplans[3] *setzt* die GeoIV "*eine Frist von fünf Jahren*[4] *für die Bereitstellung der Geobasisdaten in der durch das MGDM vorgeschriebenen Form, beginnend mit dem Datum der Existenz des MGDM*" und betont, dass "*verschiedene Rückmeldungen aus den Kantonen* zeigen*, dass die Einhaltung der geforderten Frist nicht für alle Kantone und alle Themen realistisch ist*", weshalb ein
Der "*Umsetzungsplan*"[5] wurde ausgearbeitet. Ziel dieser koordinierten Umsetzung ist es, die auf der Aggregationsinfrastruktur *geobasisdaten.ch* zur Verfügung gestellten *Geobasisdaten* zu priorisieren und dabei möglichst zeitgerecht und flächendeckend die ganze Schweiz abzudecken.

Im Abschnitt "*Laufende Umsetzungsprogramme*" des Umsetzungsplans der Konferenz der Kantonalen Ämter für Geoinformation und Landvermessung (KKG) wird der Stand der Umsetzungsprogramme dargestellt. Die Erfahrungen aus der Umsetzung des Themas Gefahrenkarten haben gezeigt, dass für die modellkonforme Umsetzung der Geobasisdaten bei der Methode des direkten Zugriffs vorgängig Anwendungsschemata bzw. logische Modelle definiert werden müssen. Aus diesem Grund haben das Koordinationsorgan für Geoinformation des Bundes (GKG) und die GKG im Rahmen eines gemeinsamen Projekts im Jahr 2016 Weisungen für den modellkonformen Austausch von Geodaten (MDX)[6] definiert. Diese Anleitung leitet die zuständigen Stellen bei der Realisierung von modellkonformen Download-Diensten an, wie sie das Geoinformationsgesetz vorschreibt.

---

[1] https://www.fedlex.admin.ch/eli/cc/2008/388/fr#art_3

[2] https://www.fedlex.admin.ch/eli/cc/2008/389/fr#art_9

[3] https://www.kgk-cgc.ch/application/files/9615/6631/6860/Umsetzungsplanung-v15_FR.pdf

[4] https://www.fedlex.admin.ch/eli/cc/2008/389/fr#art_53

[5] https://www.kgk-cgc.ch/fr/coordination/mgdm/plan-de-mise-en-oeuvre

[6] https://www.kgk-cgc.ch/application/files/2715/4762/7910/Factsheet_Roadmap_MDX_Anforderungen_an_KGDI_DE.pdf

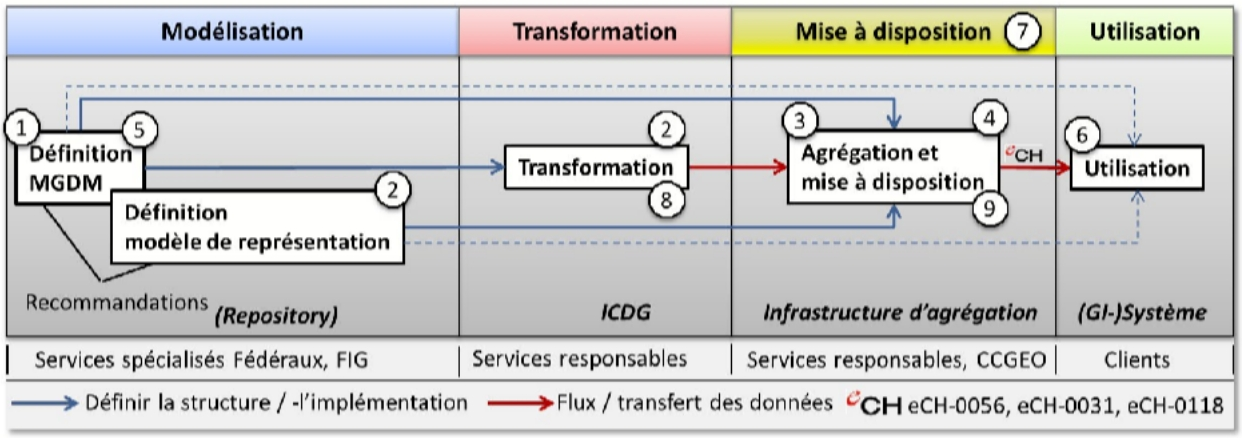

Abbildung 1: Übersicht über den Umsetzungsprozess von Geobasisdaten aus dem Umsetzungsplan für Geobasisdaten des Bundesrechts, für die die Kantone zuständig sind.

Parallel zu MDX wurde auch eine "*Empfehlung für die Entwicklung von Darstellungsmodellen*"[7] definiert, um die Kartendarstellungen in den Visualisierungsdiensten zu vereinheitlichen.

Nach ihrer Umsetzung werden die modellkonformen Geobasisdaten (MGDM) für diejenigen, die in die Zuständigkeit der Kantone fallen, auf der Aggregationsinfrastruktur der Kantone (AI) *geodienste.ch* (siehe Abbildung 1) und für diejenigen, die in die Zuständigkeit des Bundes fallen, über die Visualisierungsdienste von swisstopo auf der Bundesgeodateninfrastruktur (BGDI) veröffentlicht.

Informationen über die Konfiguration der *AI-Geodienste* sind in Berichten dokumentiert, die im Abschnitt "*Umsetzungsgrundlagen der Pilotkantone*" auf der Seite der GKG: "*Umsetzungsprozess*"[8] oder über die mit dem *geocat-Datensatz*[9] verbundenen Ressourcen, die dem *geodienste-Feed* entsprechen, eingesehen werden können.*ch* (Atom Feed / OpenSearch Service) wie im folgenden Beispiel (siehe Tabelle 1), in dem die WMS-Attribute für das MGDM Reservierte Gebiete (ID 76) detailliert aufgeführt sind .[10]

[7] https://drive.switch.ch/index.php/apps/files/?dir=/mediamaps/01-Projets/02-En-cours/01-Flache-Geodatenmod-sie/ReferenzenSfileid=6708009891#pdfviewer

[8] https://www.kgk-cgc.ch/fr/coordination/mgdm/processus-de-mise-en-oeuvre

[9] https://www.geocat.ch/geonetwork/srv/fre/catalog.search#/metadata/20288132-7b8b-471c-8239-90b1c5612c36

[10] https://www.geocat.ch/geonetwork/srv/fre/catalog.search#/metadata/a3eefbdc-1820-4fcb-bf97-9a4072af9180

Layer: Planungszonen			
Attribut Benutzerderivat	**MGDM Quelle [Attribut, Klasse]**	**WMS GetFeatureInfo**	**Bemerkung**
geometrie	Planungszone	X	Polygon
publiziert_ab	Planungszone	X	Datum
gueltig_bis	Planungszone	X	Datum
rechtsstatus	Planungszone	X	Aufzählung
bemerkung	Planungszone	X	Zeichenkette
code_typ	Typ_Planungszone	X	Text(12)
bezeichnung_typ	Typ_Planungszone	X	Text(80)
abkuerzung_typ	Typ_Planungszone	X	Text(12)
festlegung_stufe_typ	Typ_Planungszone	X	Aufzählung
bemerkung_typ	Typ_Planungszone	X	Zeichenkette
kanton		X	wird durch AI abgefüllt

Tabelle 1: Auszug aus dem Implementierungsbericht für das MGDM Reservierte Gebiete (ID 76), der die Attribute des konzeptuellen Modells angibt, die über die wms Operation GetFeatureInfo a b g e r u f e n werden müssen.

Während für die implementierten Geobasisdaten Informationen über die Konfiguration der Geodienste abgerufen werden können, gibt es für die Geodienste der BGDI keine Dokumentation.

Trotz der Anleitung für den Austausch von modellkonformen Geodaten (MDX) gibt es Unterschiede zwischen den Daten und Geodiensten, die von den Kantonen und der IA bereitgestellt werden. Dies ist insbesondere im folgenden Beispiel der Fall (siehe Abbildung 2: Unterschiede zwischen den *Geodiensten vsgis.ch* und *geodienste.ch*), in dem zwei verschiedene *Geodienste* miteinander verglichen werden, nämlich der von *geodienste.ch*[11] (rechts) und der von *vsgis.ch*[12] (links), die sich beide auf das MGDM Reservierte Gebiete (ID 76) beziehen und bei denen sowohl die Attribute als auch die Darstellung nicht übereinstimmen. Zu Demonstrationszwecken wurden beide Geodienste auf *map.geo.admin.ch* importiert und können unter folgender Adresse eingesehen werden: https://s.geo.admin.ch/bwgnkkx2ut9k.

[11] https://geodienste.ch/db/planungszonen_v1_1_0/fra?SERVICE=WMSSREQUEST=GetCapabilities

[12] https://map.vsgis.ch/valdebagnes/wsgi/mapserv_proxy?ogcserver=source+for+NoneSSERVICE=WMSSVERSION=1.3.0SRE-QUEST=GetCapabilities

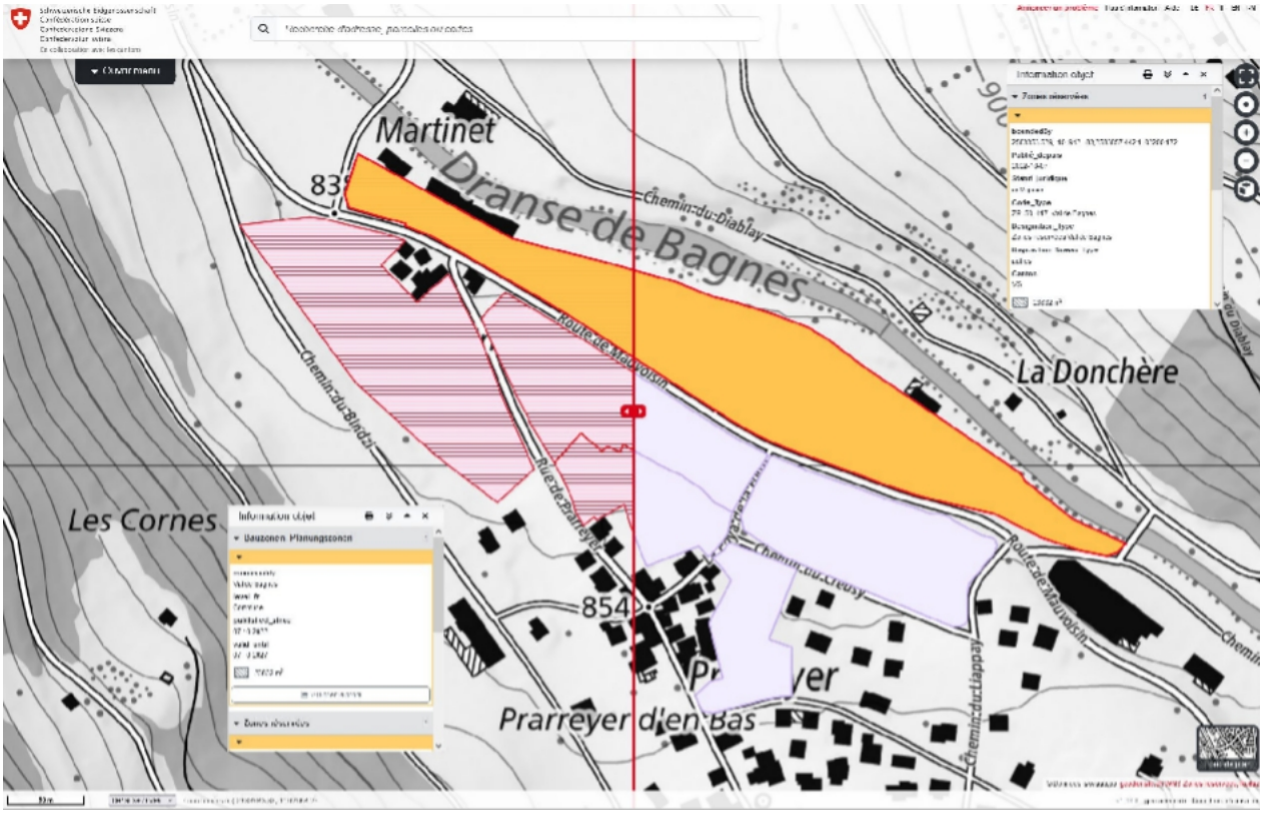

Abbildung 2: Unterschiede zwischen den Geodiensten vsgis.ch und geodienste.ch

Anm.: vsgis.ch ist zwar nicht die offizielle Publikationsinstanz der Geodaten des Kantons Wallis, hat aber den Vorteil, dass die OGC-Standards eingehalten werden und somit eine Integration in andere Instanzen wie *map.geo.admin.ch* möglich ist. Die Kohärenz mit den Daten des Kantons Wallis[13] wurde jedoch vorgängig überprüft (siehe Abbildung 3).

Fields:

- OBJECTID *(type: esriFieldTypeOID, alias: OBJECTID)*
- COMMUNE *(type: esriFieldTypeString, alias: Commune, length: 100)*
- ZR_NIVEAU_FR *(type: esriFieldTypeString, alias: Niveau, length: 50)*
- ZR_NIVEAU_DE *(type: esriFieldTypeString, alias: Festlegung Stufe, length: 50)*
- PUBLIE_DEPUIS *(type: esriFieldTypeDate, alias: Publié depuis, length: 8)*
- VALABLE_JUSQUA *(type: esriFieldTypeDate, alias: Fin validité, length: 8)*
- SHAPE *(type: esriFieldTypeGeometry, alias: SHAPE)*
- SHAPE.LEN *(type: esriFieldTypeDouble, alias: SHAPE_Length)*
- SHAPE.AREA *(type: esriFieldTypeDouble, alias: SHAPE_Area)*

Drawing Info:

Renderer:
Simple Renderer:
Symbol:
Style: esriSFSHorizontal
Color: [230, 0, 0, 255]
Outline:
Style: esriSLSSolid
Color: [230, 0, 0, 255]
Width: 2
Label: N/A
Description: N/A
Transparency: 0
Labeling Info:

Abbildung 3: Attribute und Symbologie der ArcGISOnline REST API des Kantons Wallis

[13] https://sit.vs.ch/arcgis/rest/services/PAZ/MapServer/0

Die Probleme, die sich aus diesen Feststellungen ergeben, sind die folgenden.

- zunächst für die professionellen Nutzer von Geodienstinfrastrukturen: Es gibt eine Heterogenität zwischen den Webdiensten der verschiedenen Kantone und des Bundes - sowohl bei der Veröffentlichung von Attributen als auch bei der Darstellung von Geodaten. Die Zusammenlegung mehrerer Dienste in einem einzigen Projekt kann zu Inkonsistenzen führen.

- zweitens für die Infrastrukturbetreiber von Geodiensten: Obwohl es für einige MGDMs eine Dokumentation gibt, die sowohl die Attribute, die in einen Webdienst aufgenommen werden müssen, als auch die Darstellung der Daten beschreibt, wird diese Dokumentation oft nicht gut beachtet oder ist schwer zu nutzen.

Wir können daher die Hypothese aufstellen, dass eine normativere Methode, die sogar Automatisierungen begünstigt, sowohl für die CGC (Aggregation von Daten auf geodienste) als auch für jeden Kanton oder jede Einheit, die Daten zu einem MGDM veröffentlicht, und letztlich für die professionellen Nutzer der Geodienstinfrastrukturen von Vorteil sein könnte.

In diesem Projekt sollen Wege erforscht werden, wie die geltenden Standards im Hinblick auf die Veröffentlichung von MGDM in Form von interoperablen Geodiensten optimal genutzt werden können. Dies gilt sowohl für die Definition einer denormalisierten/flachen Struktur eines minimalen Geodatenmodells als auch für eine symbologische Beschreibung.

2. Ziele des Projekts

Das Projekt **FGDM4GS** steht für "Flat Geodata Models for Geoservices" und soll die Möglichkeiten der konzeptionellen Modellierung für die Implementierung von Geodiensten auf der Grundlage von Geobasisdaten und den damit verbundenen MGDM-Anforderungen erforschen. Um die interoperable Implementierung von Geodiensten zu fördern, soll Folgendes bewertet werden:

- die Definitionsmöglichkeiten von INTERLIS VIEW für die Publikation von Attributdaten (mit Beispielen anhand ausgewählter Vorlagen)

- die grafischen Fähigkeiten SYMBOLOGY INTERLIS für die Veröffentlichung von kartografischen Darstellungen (mit Beispielen auf der Grundlage einer Auswahl von Modellen)

Am Ende jeder Analyse werden auf der Grundlage der Ergebnisse Perspektiven und Empfehlungen erstellt, die als Grundlage für eine mögliche Änderung der normativen Grundlagen und der damit verbundenen Werkzeuge dienen können. Dementsprechend sind die Adressaten der Ergebnisse dieses Projekts und der Empfehlungen die CGC/KGK und COSIG/KOGIS, sowie die eCH-Gruppen Geoinformation und OSIG/SOGI GGMM.

3. Auswahl von Geobasisdaten

3.1. Methodik

In dieser Vorphase geht es darum, Geobasisdaten auszuwählen, um zu versuchen, die Konzepte, die in dieser Arbeit bewertet werden, aus der Sicht der Denormalisierung und der Symbologie zu veranschaulichen. Dazu werden wir Modelle auswählen, für die eine Verbindung zu einem bestehenden Geodienst hergestellt werden kann, um die Kohärenz mit den implementierten Daten- und Darstellungsmodellen zu bewerten. Die über den Geodienst erhaltenen Informationen ermöglichen es auch, alle Informationen für die Definition der VIEW und des Darstellungsmodells zu erhalten.

Diese Auswahl basiert auf einer Analyse anhand der Einträge in geobasisdaten.ch, um festzustellen, inwieweit die Geobasisdaten :

- ... besitzen ein konzeptionelles Datenmodell
- ... enthalten das syntaktische Element "VIEW" der INTERLIS-Sprache, mit dem Ansichten beschrieben werden können, wie in Kapitel 4 näher ausgeführt wird
- ... sich an die Definition von Darstellungsmodellen halten, wie sie in den Re-Empfehlungen zur Entwicklung von Darstellungsmodellen für MDGM empfohlen werden[14]
- ... verbinden MGDM mit Geodiensten
    - um die Kohärenz ihrer Implementierung mit den konzeptuellen Modellen und den Darstellungsinformationen zu validieren
    - oder andernfalls zusätzliche Informationen für die Definition von VIEW und Darstellungsmodellen zu erhalten

Dazu wird eine Querverwendung der API *geobasisdaten.ch*[15] und des CSW-Dienstes[16] von *geo- cat.ch* [17] durchgeführt. Die folgende Abbildung veranschaulicht das Auswahlkonzept von MGDM.

Abbildung 4: Konzept zur Verknüpfung von MGDMs mit Geodiensten

[14] https://www.kgk-cgc.ch/fr/coordination/mgdm/modelisation-des-mgdm

[15] https://geobasisdaten.ch/api/v1/redoc/

[16] https://www.geocat.ch/geonetwork/srv/fre/csw?service=CSWSversion=2.0.2Srequest=GetCapabilities

[17] https://www.geocat.ch/geonetwork/srv/fre/catalog.search#/home

Die Informationen von *geobasisdaten.ch* werden mit den zugehörigen Geodiensten verknüpft, indem der CSW-Dienst von *geocat.ch* um die Attribute ersucht wird, die in die VIEWS integriert werden müssen, um sie angemessen konfigurieren zu können :

- Die GetFeatureInfo-Abfrage auf WMS-Diensten
- Die kartografische Darstellung sowie die Operation GetStyles

Die Analyse wurde mithilfe verschiedener Skripte durchgeführt, die über dieses GitHub-Repository[18] zugänglich sind (siehe Auswahlverfahren im Anhang).

3.2. Inventar der ausgewählten Modelle

Die API *geobasisdaten.ch* enthielt am 14.02.2023 347 Einträge, von denen 80%, d.h. 283/347 Einträge, einen Link zu einem Datenmodell (.ili) enthielten:

- Keine der vorhandenen Vorlagen enthält "VIEW". Dies bestätigt, dass dies nicht üblich ist.
- Weniger als 5% oder etwa 10/347 API-Einträge enthielten einen Link zu Darstellungsinformationen im empfohlenen tabellarischen Format. 33% oder 115/347 Einträge konnten mit Geodiensten verknüpft werden.

Schließlich wird eine Auswahl von MGDMs getroffen, wobei die oben genannten Kriterien und die interne Komplexität der Modelle berücksichtigt werden.

Daraus ergeben sich die Daten in der folgenden Tabelle, auf die wir uns stützen werden. Sie setzt die MGDMs mit den Datenmodellen sowie den dazugehörigen Geodiensten in Beziehung. Die Modelle enthalten alle Attribute, die für die Operation GetFeatureIn- fo erforderlich sind, aber diese Attribute sind in verschiedene Klassen unterteilt. Die Operation Get- Styles wird nicht berücksichtigt, da die Antwort nicht zufriedenstellend ist.

Beachten Sie, dass die Datensätze der API geobasisdaten.ch einen Link zu geocat enthalten, aber nur zu Ressourcen des Typs: "Datenmodell", wie im folgenden Beispiel (siehe Abbildung 5) für die Geobasisdaten "Schutzgebiete".

Man kann auch feststellen, dass die Plattform geocat.ch keine Ressourcen vom Typ : "Datenmodell" auf Ressourcen vom Typ "Dienst".

Dasselbe gilt für den *Geodienst geodienste.ch*, der keine Informationen zum Datenmodell, *zu geobasisdaten.ch* oder zu den Metadaten *geocat.ch* enthält.

[18] https://github.com/MediaComem/FGDM4GS/tree/main/WP1/T1-1

Name	Modell	WMS GetFeatureInfo	Bericht
Nationalstraßen-Achsen (86.1)[19]	Link[20]	Link[21]	-
Bundesinventar der historischen Verkehrswege (16.1[22] , 17.1)[23]	Link[24]	Link[25]	-
Reservierte Gebiete (76.1)[26]	Link[27]	Link[28]	Link[29]
Sachplan Verkehr Teil Straße (72.1)[30]	Link[31]	Link[32]	-

Tabelle 2: Inventar der ausgewählten Modelle

[19] https://geobasisdaten.ch/?search=86.1

[20] http://models.geo.admin.ch/ASTRA/Axis_V1_1.ili

[21] https://wms.geo.admin.ch/?SERVICE=WMSSVERSION=1.3.0SREQUEST=GetFeatureInfoSLAYERS=ch.astra.nationalstrassenach-senSQUERY_LAYERS=ch.astra.nationalstrassenach-senSCRS=EPSG:2056SBBOX=2531423.89,1155079.22,2532223.89,1155679.22SFEATURE_COUNT=1SHEIGHT=2SWIDTH=2SFOR-MAT=image/pngSINFO_FORMAT=text/plainSI=1SJ=1Slang=de

[22] https://geobasisdaten.ch/?search=16.1

[23] https://geobasisdaten.ch/?search=17.1

[24] https://models.geo.admin.ch/ASTRA/IVS_V2_1.ili

[25] https://wms.geo.admin.ch/?SERVICE=WMSSVERSION=1.3.0SREQUEST=GetFeatureInfoSLAYERS=ch.astra.ivs-natSQUERY_LAY-ERS=ch.astra.ivs-natSCRS=EPSG:2056SBBOX=2531423.89,1155079.22,2532223.89,1155679.22SFEA-TURE_COUNT=1SHEIGHT=2SWIDTH=2SFORMAT=image/pngSINFO_FORMAT=text/xmlSI=1SJ=1

[26] https://geobasisdaten.ch/?search=76.1

[27] https://models.geo.admin.ch/ARE/Zones_reservees_V1_1.ili

[28] https://geodienste.ch/db/planungszonen_v1_0_0/fra?SERVICE=WMSSVERSION=1.3.0SREQUEST=GetFeature-InfoSCRS=EPSG:3857SBBOX=742465.67788010137155652,5902893.5199219873175025,751489.76544106891378 76,5913919.95374835748225451SWIDTH=667SHEIGHT=815SLAYERS=planungszoneSFORMAT=image/pngSQUERY_LAYERS=pla-nungszoneSINFO_FORMAT=application/vnd.ogc.gmlSI=500SJ=267SFEATURE_COUNT=1

[29] https://www.geocat.ch/geonetwork/srv/fre/catalog.search#/metadata/a3eefbdc-1820-4fcb-bf97-9a4072af9180

[30] https://geobasisdaten.ch/?search=72.1

[31] https://models.geo.admin.ch/ASTRA/SectoralPlanForRoadInfrastructure_V1_4.ili

[32] https://wms.geo.admin.ch/?SERVICE=WMSSVERSION=1.3.0SREQUEST=GetFeatureInfoSLAYERS=ch.astra.sachplan-infrastruktur-strasse_kraftSQUERY_LAYERS=ch.astra.sachplan-infrastruktur-strasse_kraftSCRS=EPSG:2056SBBOX=2531423.89,1155079.22,2532223.89,1155679.22SFEA-TURE_COUNT=1SHEIGHT=2SWIDTH=2SFORMAT=image/pngSINFO_FORMAT=text/plainSI=1SJ=1

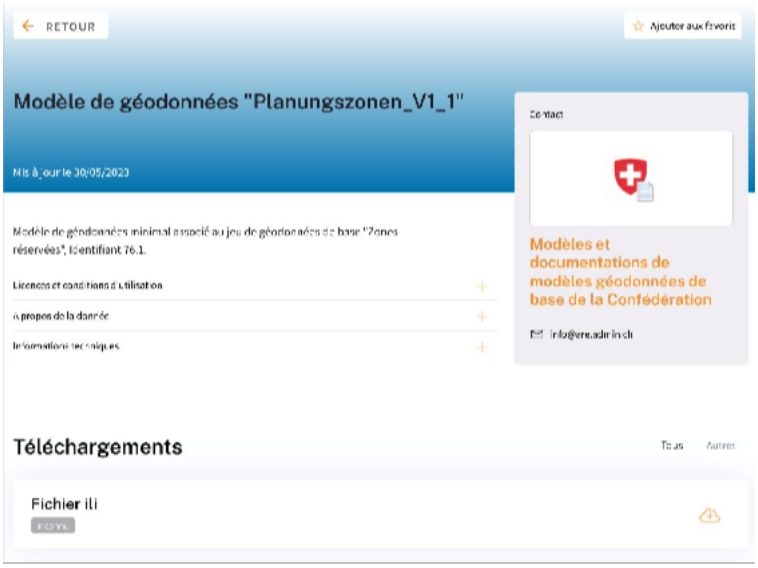

Abbildung 5: Geodatenmodell Planungszonen_V1_1[33]

3.3. Empfehlungen

3.3.1. Links zu allen Arten von Ressourcen geocat.ch

Links zu allen Arten von *geocat.ch-Ressourcen* ("Datenmodell", "Ser- vice", "Daten") auf der Ebene von *geobasisdaten.ch* integrieren, wenn verfügbar.

3.3.2. Integration von Ressourcenlinks auf geocat.ch

Integrieren Sie Links zwischen den verschiedenen Arten von *geocat.ch-Ressourcen* ("Datenmodell", "Dienst", "Daten"), wenn diese verfügbar sind.

3.3.3. Verweise auf Daten, Geodienste und Metadaten in den Modellen

Die *.ili-Modelle enthalten keine Verweise auf die Daten, die zugehörigen Geodienste oder die Metadaten. Referenzen könnten jedoch über ein Meta-Attribut oder eine spezifische Klasse oder Struktur direkt in das Modell eingefügt werden, wie es in den Rahmenmodellen[34] des ÖREB-Katasters der Fall ist.

3.3.4. Integration von Metadaten in Dienste

Die Geodienste und Daten enthalten weder einen Verweis auf Modelle noch auf Metadaten. Diese Informationen könnten den Anforderungen von eCH-0056 hinzugefügt werden.

[33] https://www.geocat.ch/datahub/dataset/d6456436-0254-4bef-b04c-df8dd1417d9c

[34] https://models.geo.admin.ch/V_D/OeREB/

4. Definition von Ansichten im konzeptuellen Modell

Art. 10[35] der GeoIV erwähnt, dass die Beschreibungssprache der GDMM einer anerkannten Norm entsprechen muss. Art. 5[36] der GeoIV swisstopo hingegen besagt, dass die allgemeine Sprache zur Beschreibung von Geodatenmodellen insbesondere dem Standard eCH-0031 IN- TERLIS 2 - Referenzhandbuch entsprechen muss[37] . Um das erste Ziel dieses Projekts zu erfüllen, evaluieren wir also die Möglichkeiten zur Erstellung von Ansichten mit INTERLIS.

Eine "VIEW" ist eine Regel in der Grammatik der INTERLIS-Sprache, die es ermöglicht, Attribute aus Klassen oder Assoziationen von Datenmodellen auszuwählen. Das Konzept der "VIEW" wird im INTERLIS-Referenzhandbuch (refhb24) im Abschnitt "2.15. Sichten" definiert. Wir beziehen uns hier auf die deutsche Version des INTERLIS-Referenzhandbuchs (refhb24), da es eine Version im asciidoc-Format gibt, die auf einem GitHub-Repository[38] veröffentlicht ist, das mit dem Konto von Geostandards.ch verknüpft ist. Dies ermöglicht es, auf die einzelnen Abschnitte des Handbuchs zu verweisen.

```
ViewDef = 'VIEW' View-Name
            Properties<ABSTRACT,EXTENDED,FINAL,TRANSIENT>
            [ FormationDef | 'EXTENDS' ViewRef ]
                { BaseExtensionDef }
                { Selection }
            '='
            [ ViewAttributes ]
            { ConstraintDef }
          'END' View-Name ';'.
ViewRef = [ Model-Name '.' [ Topic-Name '.' ] ] View-Name.
```

Codeblock 1: Auszug aus der INTERLIS BNF-Grammatik zur Definition einer VIEW

Um die grundlegenden Vorlagen nicht zu verändern, sind die VIEW-Beispiele in den folgenden Abschnitten in abgeleiteten Vorlagen definiert, d. h. in Vorlagen, die von bestehenden Vorlagen abhängen.

4.1. Methodik

In den folgenden Abschnitten wird untersucht, inwiefern "VIEWs" bei der Konfiguration von modellkonformen Geodatensätzen oder Geodiensten hilfreich sein können. Dazu nutzen wir die Dokumentation in Abschnitt 3.15 "Views" des Referenzhandbuchs und die Testbeispiele von ili2c[39] , um das Konzept der VIEW-Erstellung zu verstehen und hier näher zu erläutern. In einem zweiten Schritt werden die VIEWs auf die ausgewählten MGDMs angewendet und die Empfehlungen, die sich aus unseren Beobachtungen ergeben, werden formuliert.

---

[35] https://www.fedlex.admin.ch/eli/cc/2008/389/fr#art_10

[36] https://www.fedlex.admin.ch/eli/cc/2008/390/fr#art_5

[37] https://www.ech.ch/fr/ech/ech-0031/2.0

[38] https://github.com/geostandards-ch/doc_refhb24

[39] https://github.com/claeis/ili2c/tree/c506ae466333d726b885ca7fae4ce6825e94176d/test/data/ili23/view

4.2. Arten von VIEW

Im Folgenden werden die fünf Möglichkeiten beschrieben, VIEWS in INTERLIS zu definieren:

- PROJEKTION OF
- JOIN VON S WOBEI
- UNION OF
- AGGREGATION OF
- INSPECTION OF

4.2.1. VIEW PROJECTION OF

Laut Referenzhandbuch ist die Projektionsansicht (Schlüsselwort PROJECTION OF) *die einfachste* Ansicht. *Sie ermöglicht es, das Basisobjekt im UML-Klassendiagramm (Klasse, Struktur, Assoziation oder Ansicht) in einer modifizierten Form anzuzeigen (z. B. nur einen Teil der Attribute und in einer geänderten Reihenfolge)*. Dies kann nützlich sein, um die Anzeige von Daten zu vereinfachen oder sie an einen bestimmten Kontext anzupassen. Um beispielsweise Ansichten mit übersetzten Attributbezeichnungen bereitzustellen (z. B. eine Ansicht mit deutschen Attributnamen und eine Ansicht mit französischen Attributen), können Sie mithilfe des Konzepts "PROJECTION OF" eine Ansicht pro Sprache erstellen.

```
INTERLIS 2.3;
CONTRACTED MODEL Test AT "http://www.interlis.ch/ili2c/tests/" VERSION "1"=
  FUNCTION makeConstant(text: TEXT):TEXT;
  TOPIC Basis =
    STRUKTUR A =
      Attr1: TEXT*20;
    END A;
    CLASS B =
      Attr1: TEXT*20;
      Attr2: BAG OF A;
    UNIQUE
      Attr1;
    END B;
  END Basis;
  TOPIC ViewProjection =
    DEPENDS ON Test.Base;
    VIEW VB PROJECTION OF Test.Base.B;
     =
      ATTRIBUTE
        ALL OF B;
        Attr3 : TEXT*80 := makeConstant("hello World");
    END VB;
  END ViewProjection;
END Test.
```

Codeblock 2: Beispiel[40] für die Definition einer VIEW PROJECTION OF

[40] https://github.com/claeis/ili2c/blob/c506ae466333d726b885ca7fae4ce6825e94176d/test/data/ili23/view/view_AcceptBasicView-

4.2.2. VIEW JOIN OF & WHERE

Nach dem Referenzhandbuch (refhb24) wird mit der Verbindungsansicht (Schlüsselwort JOIN OF) das kartesische Produkt (oder Vektorprodukt) der Basisklassen (Klasse oder Ansicht) gebildet, d. h. es gibt dann so viele Objekte der Verbindungsklasse, wie es Kombinationen von Objekten der verschiedenen Basisklassen gibt. Durch die Verknüpfung mit einer WHERE-Klausel können die Join-Felder definiert werden (die über das Formationsgesetz definierte Menge an Objekt-Ansichten kann durch die Einführung von Bedingungen weiter eingeschränkt werden). Dies kann z. B. nützlich sein, um kantonale Abstimmungsstatistiken nach verschiedenen Kriterien zu verknüpfen.

```
INTERLIS 2.3;
MODEL Test AT "http://www.interlis.ch/ili2c/tests/" VERSION "1"=
  TOPIC Base =
    CLASS B =
      Attr1: TEXT*20;
      Attr2: TEXT*10;
    END B;
    KLASSE C =
      Attr3: TEXT*30;
    END C;
  END Base;
  TOPIC Join =
    DEPENDS ON Test.Base;
    VIEW BC
      JOIN OF B ~ Test.Base.B,C ~ Test.Base.C (OR NULL);
      WHERE B->Attr1 == C->Attr3;
    =
      ATTRIBUTE
        ALL OF B;
        ALL OF C;
    END BC;
  END Join;
END Test.
```

Codeblock 3: Beispiel[41] für die Definition einer VIEW JOIN OF & WHERE

4.2.3. VIEW UNION OF

Laut Referenzhandbuch (refhb24) *ermöglicht* die Ansicht union (Schlüsselwort UNION OF) *das Zusammenführen verschiedener Basisklassen zu einer einzigen Klasse. Die Attribute der verschiedenen Basisklassen werden in der Regel einem Attribut der Unionsansicht zugewiesen. Der Attributtyp der Basisklasse muss mit dem Attributtyp der Unionsansicht kompatibel sein (gleicher Typ oder Erweiterung desselben)*. Dies ermöglicht beispielsweise das Zusammenführen von 2 Datensätzen, die dieselbe Struktur besitzen.

ProjectionDef.ili

[41] https://github.com/claeis/ili2c/blob/c506ae466333d726b885ca7fae4ce6825e94176d/test/data/ili23/view/view_AcceptBasicJoinDef.ili

```
INTERLIS 2.3;
MODEL Test AT "http://www.interlis.ch/ili2c/tests/" VERSION "1"=
  TOPIC Base =
    CLASS C1 =
      Attr1: TEXT*10;
    END C1;
    CLASS C2 =
      Attr2: TEXT*30;
    END C2;
  END Base;
  TOPIC Union =
    DEPENDS ON Test.Base;
    VIEW CC
      UNION OF C1 ~ Test.Base.C1,C2 ~ Test.Base.C2;
    =
      ATTRIBUTE
        Attr1 : TEXT*30 := C1->Attr1,C2->Attr2;
    END CC;
  END Union;
END Test.
```

Codeblock 4:Beispiel[42] für die Definition einer VIEW UNION OF

4.2.4. VIEW AGGREGATION OF

Laut Referenzhandbuch (refhb24) *können* Sie mit der Aggregationsansicht (Schlüsselwort AGGREGATION OF) *alle Instanzen eines Basissatzes oder alle Instanzen, die mit der gewünschten Attributkombination identisch sind, zu einer einzigen Instanz zusammenfassen*. Auf diese Weise können Sie beispielsweise Statistiken über einen Datensatz berechnen.

```
INTERLIS 2.3;
CONTRACTED MODEL Test AT "http://www.interlis.ch/ili2c/tests/" VERSION "1"=
  TOPIC Base =
    CLASS B =
      Attr1: TEXT*20;
    END B;
  END Basis;
  FUNCTION countB(elements : BAG OF Test.Base.B):NUMERIC;
  TOPIC Aggregation =
    DEPENDS ON Test.Base;
    VIEW VB2
      AGGREGATION OF Test.Base.B EQUAL(B->Attr1);
    =
      ATTRIBUTE
```

[42] https://github.com/claeis/ili2c/blob/c506ae466333d726b885ca7fae4ce6825e94176d/test/data/ili23/view/view_AcceptBasicUnionDef.ili

```
        ElementCount : 0 ... 10000 := countB(AGGREGATES);
    END VB2;
  END Aggregation;
END Test.
```

Codeblock 5: Beispiel[43] für die Definition einer VIEW AGGREGATION OF

4.2.5. VIEW INSPECTION OF

Laut Referenzhandbuch (refhb24) *kann* man mit der Inspektionsansicht (Schlüsselwort INSPECTION OF) *die Gesamtheit aller strukturierten Elemente (definiert über BAG OF, LIST OF oder in Übereinstimmung mit einer Linie, einer einfachen Fläche oder einer Partition eines Gebiets) abrufen, die zu einem (direkten oder indirekten) Substrukturattribut einer Objektklasse gehören*. Dies ermöglicht es, eine Liste von Elementen zu sprengen, um sie strukturiert anzuzeigen.

```
INTERLIS 2.3;
MODEL Test AT "http://www.interlis.ch/ili2c/tests/" VERSION "1" =
  TOPIC Base =
    STRUKTUR A =
      Attr1: TEXT*20;
    END A;
    CLASS B =
      Attr2: BAG OF A;
    END B;
    VIEW VB
      INSPECTION OF Test.Base.B->Attr2;
    =
      ATTRIBUTE
      Attr1: TEXT*20 := B->Attr1;
    END VB;
  END Basis;
END Test.
```

Codeblock 6: Beispiel[44] für die Definition einer VIEW INSPECTION OF

4.3. Beispiele auf der Grundlage der ausgewählten Vorlagen

Die folgenden Beispiele von VIEWS für die ausgewählten Datenmodelle wurden in INTERLIS erstellt. Sie illustrieren die Möglichkeiten, die INTERLIS für die Definition von VIEWS bietet, und beziehen sich auf die folgenden Datenmodelle: `Axis_LV95_V1_1`, `IVS_V2_1`, `Planungszonen_V1_1`, `BaseModel_SectoralPlans_LV95_V1_4`.

[43] https://github.com/claeis/ili2c/blob/c506ae466333d726b885ca7fae4ce6825e94176d/test/data/ili23/view/view_AcceptBasicAggregationDef.ili

[44] https://github.com/claeis/ili2c/blob/c506ae466333d726b885ca7fae4ce6825e94176d/test/data/ili23/view/view_AcceptBasenameOfUnrenamedInspectionDef.ili

4.3.1. Beispiel 1: Achsen von Nationalstraßen

Das 1er Beispiel wird auf der Grundlage des MGDM Achsen der Nationalstraßen (86.1[45]) definiert.

Die Definition der VIEW mobilisiert drei verschiedene Klassen durch die Kombination der JOIN OF-Ansicht und die Verwendung der WHERE-Klausel. Um das grafische Modell in Abschnitt 5 definieren zu können, muss das zusätzliche Attribut "Type" verwendet werden, das dem Attribut "AxisType" der Klasse "Axis" entspricht.

```
INTERLIS 2.3;
MODEL Axis_LV95_V1_1_d (en)
  AT "mailto:maxime.collombin@heig-vd.ch"
  VERSION "2023-12-06" =
    IMPORTE Axis_LV95_V1_1;
    TOPIC Axis_d EXTENDS Axis_LV95_V1_1.Axis =
      VIEW view_Axis
        JOIN OF Axis_LV95_V1_1.Axis.Axis, Axis_LV95_V1_1.Axis.AxisSegment,
        Axis_LV95_V1_1.Axis.Sector;
        WHERE
          Axis->rAxisSegment == AxisSegment
          AND
          AxisSegment->rLinearReference == Sector;
          =
        ATTRIBUTE
          wkb_geometry := AxisSegment -> Geometry;
          Eigenumer := Axis -> Owner;
          Segmentname := AxisSegment -> SegmentName;
          Strassennummer := Axis -> AxisName;
          Bezeichnung := Axis -> AxisNameLong;
          Positionscode := Axis -> AxisPositionCode;
          Type := Axis -> AxisType;
          Bezugspunkt_Name := Sector -> SectorName;
          Kilometerwert_km := Sector -> Km;
          Sektorlange_m := Sector -> SectorLength;
          Sequenz := Sector -> Sequence;
      END view_Axis;
    END Axis_d;
END   Axis_LV95_V1_1_d.
```

Codeblock 7: abgeleitete Vorlage[46] der Vorlage Axis_LV95_V1_1 mit der Definition der VIEW

[45] https://geobasisdaten.ch/?search=86.1

[46] https://github.com/MediaComem/FGDM4GS/blob/main/WP2-WP3/model/Axis_V1_1/Axis_V1_1_d.ili

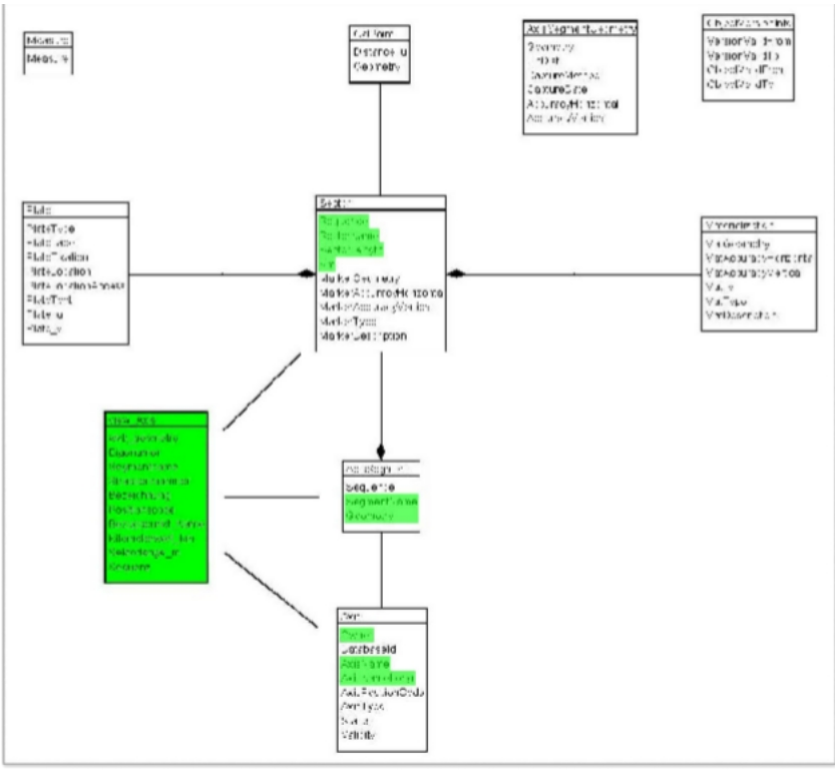

Abbildung 6: UML-Klassendiagramm mit den in der VIEW verwendeten Klassen und Attributen

4.3.2. Beispiel 2: Inventar historischer Verkehrswege

Das 2ème Beispiel ist auf der Grundlage des MGDM Bundesinventar der historischen Verkehrswege (16.1[47] , 17.1 [48]) definiert. Da das StandardSymbology-Modell[49] , das in Abschnitt 5 erläutert wird, in INTERLIS 2.4 definiert ist, müssen die Datenmodelle jedoch angepasst werden, damit sie kompatibel sind.

Die Definition der VIEW mobilisiert 7 verschiedene Klassen durch die Kombination der JOIN OF-Ansicht und die Verwendung der WHERE-Klausel mit mehreren Bedingungen.

[47] https://geobasisdaten.ch/?search=16.1

[48] https://geobasisdaten.ch/?search=17.1

[49] https://models.interlis.ch/refhb24/StandardSymbology.ili

```
INTERLIS 2.4;
MODEL IVS_V3_d
AT "mailto:maxime.collombin@heig-vd.ch"
VERSION "2023-12-04" =
    IVS_V3-IMPORTE;
    TOPIC IVS_Ik_d EXTENDS IVS_V3.IVS_Inventarkarte =
        VIEW ivs_nat
            JOIN OF IVS_V3.IVS_Inventarkarte.ivs_linienobjekte_base, IVS_V3.IVS_Inven-
tarkarte.ivs_linienobjekte_lv95, IVS_V3.IVS_Inventarkarte.ivs_objekte, IVS_V3.IVS_Inven-
tarkarte.ivs_signatur_linie, IVS_V3.IVS_Inventarkarte.ivs_kantone, IVS_V3.IVS_Inven-
tarkarte.ivs_streckenbeschriebe, IVS_V3.IVS_Inventarkarte.ivs_slanamen;
            WHERE
                ivs_slanamen->Role_ivs_objekte == ivs_linienobjekte_base
                AND
                ivs_objekte->Role_ivs_kantone == ivs_kantone
                AND
                ivs_streckenbeschriebe->Role_ivs_objekte == ivs_objekte
                AND
                ivs_linienobjekte_base->Role_ivs_signatur_linie == ivs_signatur_linie
                AND
                ivs_linienobjekte_base->Role_ivs_objekte == ivs_objekte;
                =
            ATTRIBUTE
                wkb_geometry := ivs_linienobjekte_lv95 -> ivs_geometrie;
                ivs_nummer := ivs_objekte -> ivs_nummer;
                ivs_signatur_label := ivs_signatur_linie -> ivs_deutsch;
                ivs_kanton := ivs_kantone -> ivs_kanton;
                ivs_sladatehist := ivs_streckenbeschriebe -> ivs_sladatehist;
                ivs_sladatemorph := ivs_streckenbeschriebe -> ivs_sladatemorph;
                ivs_slabedeutung := ivs_objekte -> ivs_slabedeutung;
                ivs_sortsla := ivs_objekte -> ivs_sortsla;
                ivs_slaname := ivs_slanamen -> ivs_slaname;
        END ivs_nat;
    END IVS_Ik_d;
END IVS_V3_d.
```

Codeblock 8: abgeleitete Vorlage[50] der Vorlage IVS_V2_1 mit der Definition der VIEW

[50] https://github.com/MediaComem/FGDM4GS/blob/main/WP2-WP3/model/IVS_V2_1/IVS_V3_d.ili

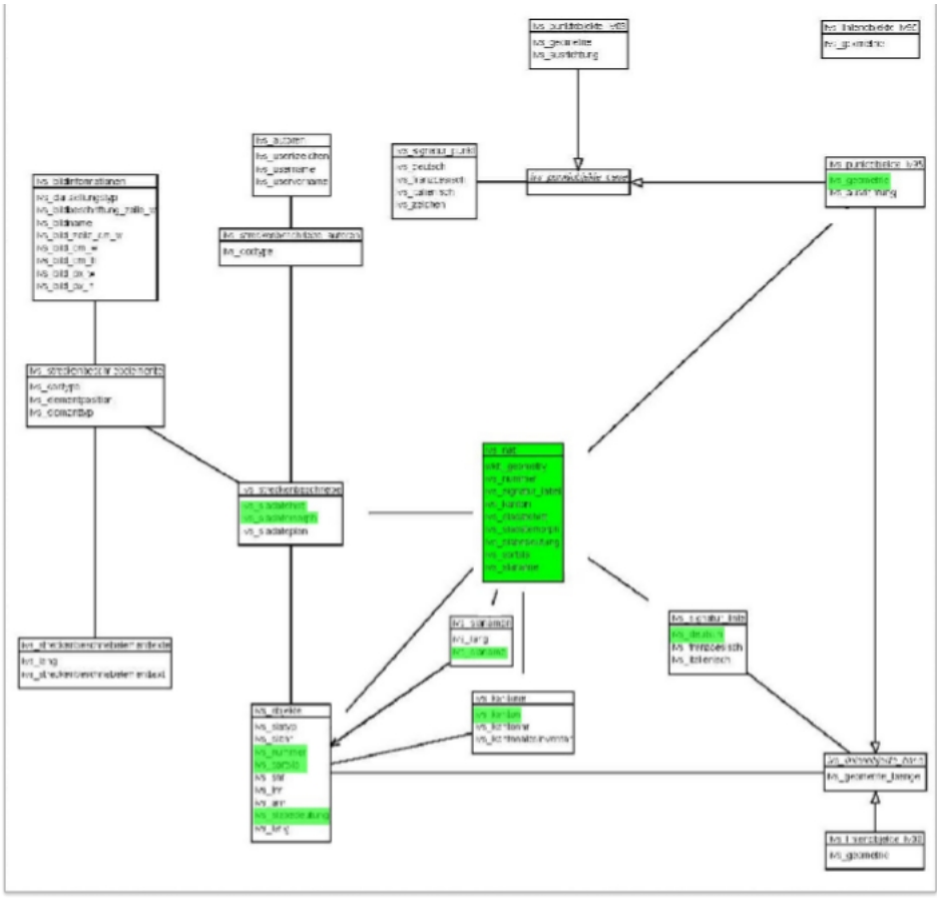

Abbildung 7: UML-Klassendiagramm mit den in der VIEW verwendeten Klassen und Attributen

4.3.3. Beispiel 3: Reservierte Bereiche

Das 3ème Beispiel wird auf der Grundlage des MGDM Reservierte Gebiete (76.1[51]) und des Implementierungsberichts[52] definiert. Da das StandardSymbology-Modell[53] , das in Abschnitt 5 erforscht wird, in INTERLIS 2.4 definiert ist, müssen die Datenmodelle dennoch angepasst werden, damit sie kompatibel sind.

Die Definition der VIEW mobilisiert 2 verschiedene Klassen durch die Kombination der JOIN OF-Ansicht und der Verwendung der WHERE-Klausel.

```
INTERLIS 2.4;
MODEL Planungszonen_V2_d_A
AT "mailto:maxime.collombin@heig-vd.ch"
VERSION "2023-12-04" =
    IMPORTS Planungszonen_V2;
    TOPIC PZ_d EXTENDS Planungszonen_V2.Geobasisdaten =
```

[51] https://geobasisdaten.ch/?search=76.1

[52] https://www.geocat.ch/geonetwork/srv/fre/catalog.search#/metadata/a3eefbdc-1820-4fcb-bf97-9a4072af9180

[53] https://models.interlis.ch/refhb24/StandardSymbology.ili

```
      VIEW view_pz
        JOIN OF Planungszonen_V2.Geobasisdaten.Planungszone, Planungszonen_V2.Geobasis-
daten.Typ_Planungszone;
        WHERE
           Planungszone->TypPZ == Typ_Planungszone;
           =
        ATTRIBUTE
         wkb_geometry := Planungszone -> Geometrie;
         publiziert_ab := Planungszone -> publiziertAb;
         gueltig_bis := Planungszone -> publiziertBis;
         rechtsstatus := Planungszone -> Rechtsstatus;
         bemerkungen := Planungszone -> Bemerkungen;
         code_typ := Typ_Planungszone -> Code;
         bezeichnung_typ := Typ_Planungszone -> Bezeichnung;
         abkuerzung_typ := Typ_Planungszone -> Abkuerzung;
         festlegung_stufe_typ := Typ_Planungszone -> Festlegung_Stufe;
         bemerkung_typ := Typ_Planungszone -> Bemerkungen;
      END view_pz;
    END PZ_d;
END Planungszonen_V2_d_A.
```

Codeblock 9: Erste abgeleitete Vorlage[54] der Vorlage Planungszonen_V2 mit der Definition von VIEW

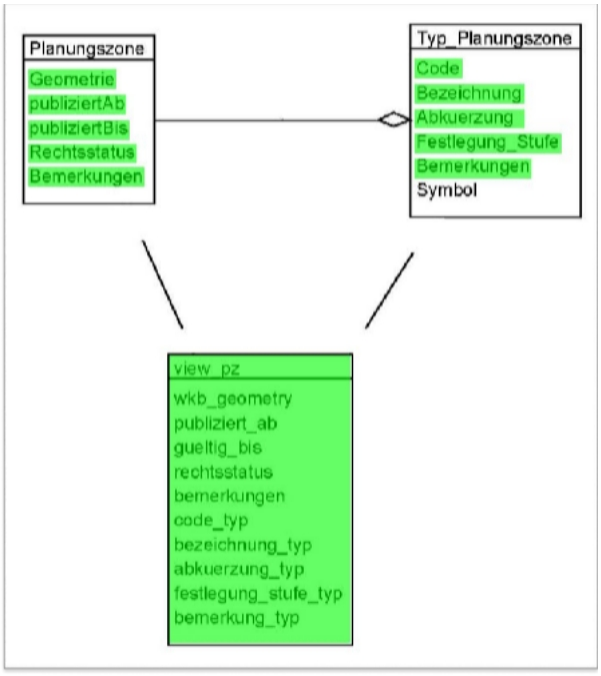

Abbildung 8: UML-Klassendiagramm mit den in der VIEW verwendeten Klassen und Attributen

[54] https://github.com/MediaComem/FGDM4GS/blob/main/WP2-WP3/model/Planungszonen_V1_1/Planungszonen_V2_d_A.ili

4.3.4. Beispiel 4: Sachplan Verkehr Teil Infrastruktur Straße

Das 4ème Beispiel ist auf der Grundlage des MGDM Sektorplan Verkehr Teil Infra- Struktur Straße (72.1[55]) definiert. Da das Modell Standard Symbology[56] , das in Abschnitt 5 erforscht wird, in INTERLIS 2.4 definiert ist, müssen die Datenmodelle dennoch angepasst werden, damit sie kompatibel sind. Da die in das Modell importierten Modelle in INTERLIS 2.3 definiert wurden, war es nicht möglich, das abgeleitete Modell zu definieren, ohne dass es zu Fehlern kommt.

Ein Attribut *Beschreibung* wird in Form eines *STRUCTURE-Elements MultilingualMText* definiert (verschachtelte Struktur = ein Attribut pro Sprache). Dies bedeutet, dass dieses Attribut *Beschreibung* entweder durch ein einzelnes Attribut dieses STRUCTURE-Elements oder durch die Konka- tation mehrerer Attribute, die jeweils eine Übersetzung darstellen, übersetzt werden könnte.

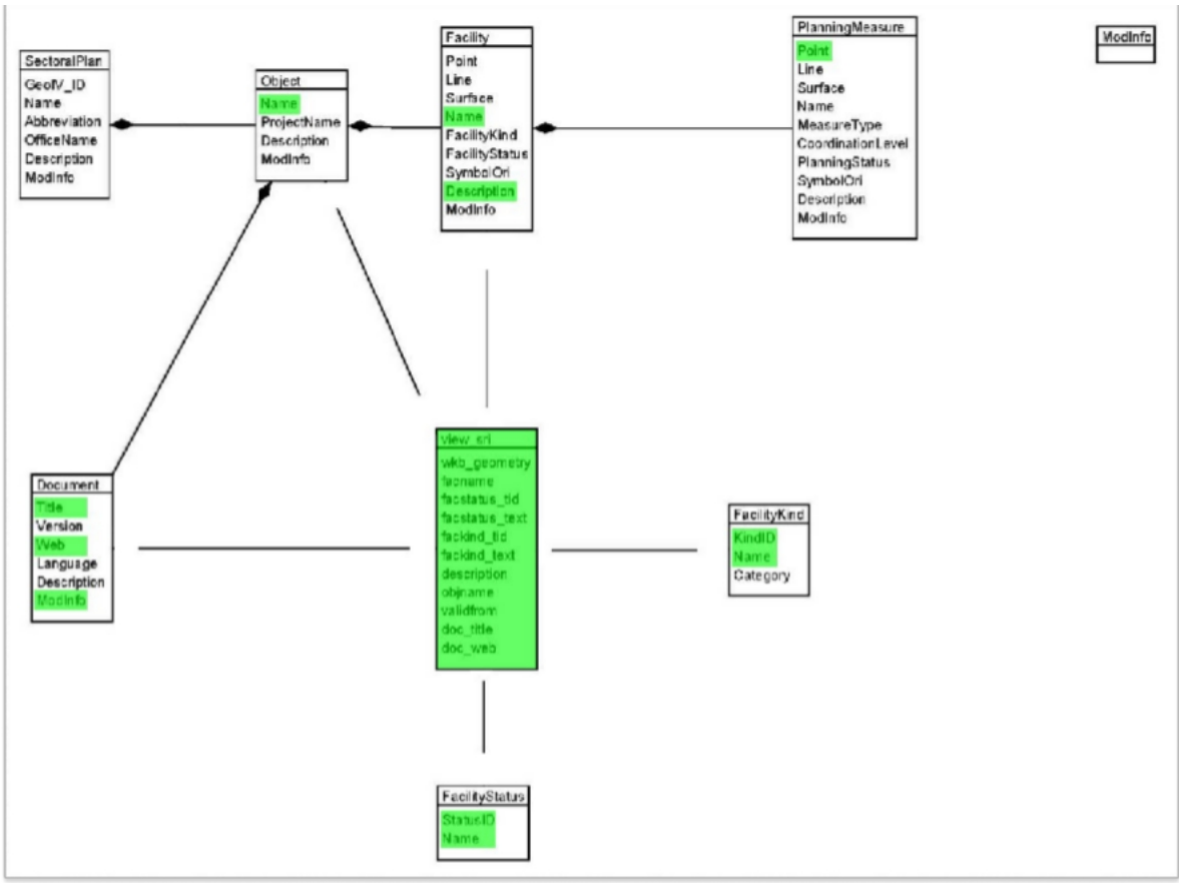

Abbildung 9: UML-Klassendiagramm mit den in der VIEW verwendeten Klassen und Attributen

[55] https://geobasisdaten.ch/?search=72.1

[56] https://models.interlis.ch/refhb24/StandardSymbology.ili

Die Definition der VIEW mobilisiert 10 verschiedene Klassen durch die Kombination der JOIN OF-Ansicht und die Verwendung der WHERE-Klausel mit mehreren Bedingungen.

```
INTERLIS 2.3;
MODEL BaseModel_SectoralPlans_LV95_V1_4_d
AT "mailto:maxime.collombin@heig-vd.ch"
VERSION "2023-12-17" =
   IMPORTS BaseModel_SectoralPlans_LV95_V1_4,
  BaseModel_SectoralPlans_Catalogues_V1_4;
     TOPIC SPRI_d=
     DEPENDS ON BaseModel_SectoralPlans_Catalogues_V1_4.Catalogue_FacilityStatus;
     DEPENDS ON
    BaseModel_SectoralPlans_LV95_V1_4.
    SectoralPlans_WithLatestModification;
    DEPENDS ON
    BaseModel_SectoralPlans_Catalogues_V1_4.Catalogue_FacilityKind;
       VIEW view_sri
        JOIN OF
          BaseModel_SectoralPlans_LV95_V1_4.
          SectoralPlans_WithLatestModification.Facility,
          BaseModel_SectoralPlans_Catalogues_V1_4.
          Katalog_FacilityStatus.FacilityStatus,
          BaseModel_SectoralPlans_Catalogues_V1_4.
          Katalog_FacilityKind.FacilityKind,
          BaseModel_SectoralPlans_LV95_V1_4.
          SectoralPlans_WithLatestModification.Object,
          BaseModel_SectoralPlans_LV95_V1_4.
          SectoralPlans_WithLatestModification.Document;
        WHERE
          Facility->Object==Object
          AND
          Document->Object==Object;
          =
        ATTRIBUTE
          wkb_geometry := Facility -> Point;
          /* wkb_geometry := Facility -> Line; andere Option für die Geometrie */
          /* wkb_geometry := Facility -> Surface; other option for the geometry
          */ facname := Facility -> Name;
          facstatus_tid := FacilityStatus -> StatusID;
          facstatus_text := FacilityStatus -> Name;
          fackind_tid := FacilityKind -> KindID;
          fackind_text := FacilityKind -> Name;
          description := Facility -> Description;
          objname := Object -> Name;
          validfrom := Document -> ModInfo; /* this attribute is ambiguous; only the attribute
                                       validfrom is needed; this definition is probably wrong */
          doc_title := Document -> Title;
```

```
          doc_web := Document -> Web;
        END view_sri;
      END SPRI_d;
END BaseModel_SectoralPlans_LV95_V1_4_d.
```

Codeblock 10: abgeleitetes Modell[57] des Modells BaseModel_SectorPlans_LV95_V1_4 mit der VIEW-Definition

4.4. Empfehlungen

4.4.1. Erstellen einer abgeleiteten Vorlage

Um die grundlegenden Modelle nicht zu verändern, schlagen wir vor, abgeleitete Modelle für die Definition von VIEWS für die Veröffentlichung von Geodiensten zu erstellen. Dies ermöglicht es, die Verantwortlichkeiten getrennt zu halten, auf der einen Seite rein auf die Modellbeschreibung, auf der anderen Seite auf die Elemente zur Veröffentlichung.

4.4.2. Ansicht vom Typ Projektion (PROJECTION OF)

Wir empfehlen die Verwendung der Projektionsansicht (Stichwort PROJECTION OF) für einfache Modelle, bei denen es darum geht, die Daten einer einzelnen Klasse zu verteilen (z. B. durch Umbenennung der Attributbezeichnungen).

4.4.3. Ansicht vom Typ Verbindung (JOIN OF)

Der Ansichtstyp Join (Schlüsselwort JOIN OF) in Verbindung mit der WHERE-Klausel ermöglicht es, Elemente aus mehreren CLASS in einer Ansicht zusammenzufassen. Dieser Typ ist für Fälle zu bevorzugen, in denen Attribute aus mehreren CLASS in einer Ansicht zusammengefasst werden.

4.4.4. ili2db-Unterstützung vom Typ Projektion und Verbindung

ili2db unterstützt keine VIEWs und ermöglicht es daher nicht, direkt Ansichten aus dem Modell zu erstellen. Es wäre daher wichtig, dass ili2db VIEWS unterstützt, um die Veröffentlichung von Geodiensten zu erleichtern. Nach bisherigen Empfehlungen könnten die Typen pro- jektion und junction die einzigen (oder vorrangigen) zu implementierenden Typen sein.

4.4.5. Normativer Ansatz für Ansichten

Die Veröffentlichung von INTERLIS-Ansichten ermöglicht es, eine denormalisierte Struktur eines INTERLIS-Modells zu definieren, um die Veröffentlichung von Geodiensten in einer einheitlichen Art und Weise für ein MGDM zu erleichtern. Es wird vorgeschlagen, einen normativen Ansatz in Gang zu setzen, z. B. :

- mit einem neuen eCH-Standard
- mit einer Anpassung der bestehenden Standards: eCH-0031 und eCH-0056

--

[57] https://github.com/MediaComem/FGDM4GS/blob/main/WP2-WP3/model/SectoralPlanForRoadInfrastructure_V1_4/SectoralPlanForRoadInfrastructure_V1_4_d.ili

5. Darstellungsmodell

Seit Juni 2014 unterstützen im Rahmen der konkreten Arbeiten zur Harmonisierung der Geobasisdaten des Bundesrechts, insbesondere zur Erarbeitung von minimalen Geodatenmodellen (MGDM), Empfehlungen die Fachinformationsgemeinschaften (FIG) bei der Erarbeitung von Darstellungsmodellen (DstM).

Im Dokument zu diesen Empfehlungen heisst es: "*Die zuständigen Fachstellen des Bundes sollen sich bei der Definition von DstM für minimale Geodatenmodelle (MGDM) insbesondere auf diese Empfehlungen stützen, um einen einheitlichen Typ der Darstellungsbeschreibung zu erhalten. Dies vereinfacht die spätere Umsetzung in den Anwendungen.*", dass es also "*in erster Linie* um *die einheitliche Darstellung* geht, *die über den Darstellungsdienst zur Verfügung gestellt werden muss*" und dass "*die Webportale, die im Rahmen der NGDI aufgebaut werden, diese DstM ebenfalls nutzen können müssen*".

Der vorgeschriebene Ansatz besteht darin, die Informationen zur grafischen Darstellung in verschiedenen möglichen Formen zu dokumentieren (siehe Abbildung 13): auf der Grundlage eines tabellarischen Modells, eines PDF-Dokuments oder sogar einer Beschreibung im OGC-Format SLD/SE. Trotz dieser Vielfalt an Möglichkeiten kann man heute feststellen, dass diese Empfehlungen sehr wenig befolgt werden, mit nur ~3% der Einträge auf *geobasisdaten.ch,* die von einer solchen Dokumen- tation begleitet werden. Wir stellen daher folgende Hypothesen auf:

- Die Verwendung von Stildefinitionen in einem objektorientierten Modell kann Verwirrung stiften und möglicherweise sogar zu Konflikten führen - z. B. wenn Stile für zwei CLASS definiert wurden, der zu implementierende Webdienst aber die Elemente der beiden CLASS in einer flachen Struktur kombinieren soll. Die Bereitstellung einer Ansicht, die eine zusammengefügte Struktur darstellt, wird dies einfacher machen.

- die Tatsache, dass die Informationen zur grafischen Darstellung in INTERLIS beschrieben werden, könnte dazu führen, dass die Modelle zur grafischen Darstellung zusammen mit den konzeptuellen Modellen auf den Modell-Repositories veröffentlicht werden und so den Prozess der Implementierung von Geodaten und Geodiensten, die auf MGDM basieren, fördern.

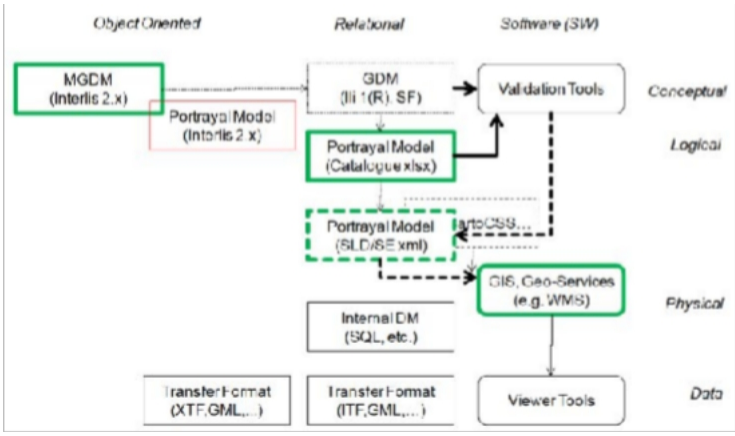

Abbildung 10: Konzept der kreuzgrafischen Darstellung INTERLIS-OGC

--

INTERLIS definiert ein grafisches Modell (.ili) und grafische Signaturen (.xtf). Die grafische Signatur (.xtf) ist mit dem grafischen Modell (.ili) verknüpft, das wiederum mit dem Datenmodell verknüpft ist. (siehe Abbildung 11: Vergleich der Konzepte für die grafische Darstellung von INTERLIS (links) und OGC (rechts)).

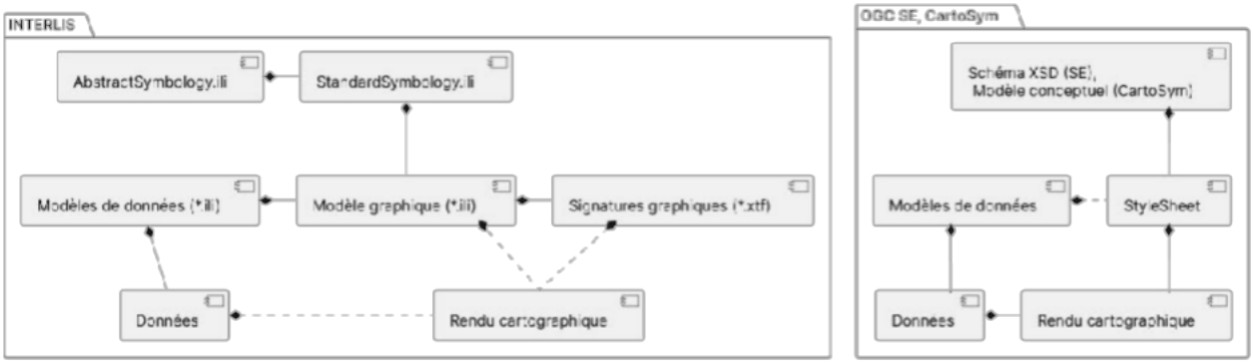

Abbildung 11: Vergleich der grafischen Darstellungskonzepte von INTERLIS (links) und OGC (rechts)

Aus Sicht des OGC ermöglichen die Standards OGC Symbology Encoding 1.1.0[58] und OGC Styled Layer Descriptor 1.1.0[59] , die in den MGDM-Empfehlungen empfohlen werden, die Beschreibung von Kartenstilen, die, wenn sie mit Datensätzen oder Geodaten verknüpft werden, eine kartografische Darstellung ermöglichen (siehe Abbildung 11: Vergleich zwischen INTERLIS (links) und OGC (rechts) Konzepten zur grafischen Darstellung). Während diese Standards durch XML Schema modelliert und auf die XML-Kodierung beschränkt sind, hat das OGC im Jahr 2020 den Standard OGC Symbology Conceptual Model: Core Part[60] veröffentlicht, der ein modulares und erweiterbares konzeptuelles Modell beschreibt, das die Möglichkeit bietet, mehrere Kodierungen zu verwenden, um eine größere Interoperabilität der Kartenstile zu fördern. Es wird derzeit in Richtung des zukünftigen OGC-Standards Cartographic Symbology (Carto- Sym) weiterentwickelt und ergänzt den konzeptionellen Ansatz durch einen Mehrparteienansatz mit zwei grundlegenden Kodierungen: Cartographic Symbology in CSS (CartoSymCSS) und Cartogra- phic Symbology in JSON (CartoSymJSON).

5.1. Methodik

Die grafischen Darstellungsmöglichkeiten von INTERLIS werden im Hinblick auf die Standards des Open Geospatial Consortium (OGC) analysiert und die ausgewählten MGDMs werden beispielhaft dargestellt - alles mit dem Ziel, herauszufinden, inwiefern die grafische Beschreibung in INTERLIS in den Implementierungsprozess der MGDMs integriert werden könnte.

In den folgenden Abschnitten analysieren wir die Fähigkeiten zur grafischen Darstellung auf der Grundlage der Anforderungsklassen von CartoSym und schlagen Beispiele für grafische Signaturen vor. Schließlich werden Empfehlungen in Bezug auf unsere Beobachtungen formuliert.

[58] https://portal.ogc.org/files/?artifact_id=16700

[59] https://portal.ogc.org/files/?artifact_id=22364

[60] https://docs.ogc.org/is/18-067r3/18-067r3.html

5.2. Vergleich INTERLIS und OGC

Gemäss Abschnitt 3.16 "Graphische Darstellungen" des INTERLIS-Referenzhandbuchs besteht eine graphische Darstellung aus graphischen Definitionen ("GraphicDef"), die immer auf einer Ansicht oder einer Klasse basieren (Stichwort "BASED ON"). Auf konzeptioneller Ebene hat eine Graphikdefinition den Zweck, jedem Objekt dieser Ansicht oder Klasse, das sich auf die Ansicht oder Klasse bezieht, durch eine oder mehrere Zeichnungsregeln ("DrawingRule") eine eindeutige Graphiksignatur (Punktsymbol, Linie, Flächenfüllung, Label) zuzuweisen und dabei ein oder mehrere Graphikobjekte zu erzeugen, die die entsprechende Darstellung nach sich ziehen.

Die einzigartigen grafischen Signaturen: "PolylineSign", "Symbolsign", "SurfaceSign", "TextSign", die in Transferdateien definiert sind, entsprechen "Line- Symbolizer", "PointSymbolizer", "PolygonSymbolizer" bzw. "TextSymbolizer" im Kontext von OGC SE. Die Eigenschaften von CartoSym sind in den Anforderungsklassen "Basic Vector Features Styling", "Basic Labeling", "Advanced strokes" und "Advanced fills" definiert.

Die Rasterdarstellung ("RasterSymbolizer" eines "CoverageStyle" für OGC SE oder Anforderungsklasse "Coverage Styling" für CartoSym) wird vom INTERLIS Symbologie-Modell nicht unterstützt.

Im Kontext von OGC SLD/SE besteht ein FeatureTypeStyle aus einer oder mehreren Regeln ("Rule"), die wiederum aus 0 bis n Filtern ("Filter") und 0 bis n Symbolisierern ("Symbolizer") bestehen. Symbolisierer können vom Typ Linie ("LineSymboli- zer"), Punkt ("PointSymbolizer"), Polygon ("PolygonSymbolizer") oder Text ("TextSym- bolizer") sein.

Was CartoSym betrifft, so ist die Spezifikation in Anforderungsklassen definiert, nämlich insbesondere Core[61] , Basic Vector Features Styling[62] , Basic Labelling[63] , Advanced strokes[64] , Advanced fills[65] und die für Coverage Styling .[66]

---

61 https://docs.ogc.org/DRAFTS/18-067r4.html#rc-core

62 https://docs.ogc.org/DRAFTS/18-067r4.html#rc-vector

63 https://docs.ogc.org/DRAFTS/18-067r4.html#_requirements_classes_for_labeling

64 https://docs.ogc.org/DRAFTS/18-067r4.html#_requirements_classes_for_advanced_strokes

65 https://docs.ogc.org/DRAFTS/18-067r4.html#_requirements_classes_for_advanced_fills

66 https://docs.ogc.org/DRAFTS/18-067r4.html#rc-coverage

CartoSym	OGC SE	INTERLIS
Core (Selector)	<ogc:Filter>	Modèle Graphique
Basic Vector Features Styling, Basic Labeling, Advanced strokes, Advanced fills	FeatureTypeStyle (<LineSymbolizer>, <PointSymbolizer>, <PolygonSymbolizer>, <TextSymbolizer>)	Signature Graphique (PolylineSign, Symbolsign, SurfaceSign, TextSign)
Coverage styling	CoverageStyle <RasterSymbolizer>	N/A

Tabelle 3: CartoSym-Anforderungsklassen und SE- und INTERLIS-Äquivalente

Der folgende Abschnitt führt einen Vergleich zwischen INTERLIS OGC SLD/SE und CartoSym unter Berücksichtigung der Anforderungsklassen von CartoSym durch.

5.3. Anforderungsklasse "Kern"

Die in diesem Abschnitt beschriebenen Anforderungen definieren die Core-Anforderungsklasse der Symbologie. Der Vergleich der Elemente der Anforderungsklasse "Core" wird in den folgenden Abschnitten beschrieben.

5.3.1. Metadaten

Laut CartoSym enthalten Metadaten beschreibende Informationen über ein "style" oder "stylesheet". Sie können einen Titel, eine Zusammenfassung, eine Beschreibung, eine Liste von Autoren, eine Liste von Schlüsselwörtern und eine Liste von Geodatenklassen enthalten. Allen Metadatenelementen wird ein Punkt (.) als erstes Zeichen einer Zeile vorangestellt. Es gibt kein eigentliches Äquivalent in der INTERLIS-Syntax.

5.3.2. Stile

Laut CartoSym entspricht ein "Style" oder "Stylesheet" der höchsten hierarchischen Ebene des Konzepts der kartografischen Darstellung. Ein Stil besteht aus einer "StylingRule" und Metadaten ("metadata"). Im Kontext von INTERLIS entspricht ein Stil einem "GRAPHIC".

5.3.3. Styling-Regeln

Laut CartoSym legt eine "StylingRule" fest, dass ein bestimmter Symbolizer (der definiert, wie Daten dargestellt werden) angewendet werden soll, wenn die Regel auf der Grundlage eines "Selector"-Ausdrucks ausgewählt wird (z. B. Bedingungen für Eigenschaften von Entitäten oder Kartenmaßstäbe). Eine "StylingRule" kann verschachtelte Regeln (neste- dRules) enthalten, die nur dann angewendet werden, wenn der "Selector" der übergeordneten Regel auf "*true*" gesetzt ist, und deren Symbolizer den "Symbolizer" der übergeordneten "StylingRule" erben und ersetzen.

Der Vergleich zwischen den Stilregeln ("StylingRule") und den Elementen des INTERLIS-Symbologiemodells wird in der folgenden Tabelle beschrieben.

CartoSym	SE	INTERLIS
name	<se:Name> inside <se:Rule>	Symbol Of inside DrawingRule inside GraphicDef
Symbolizer	<se:PointSymbolizer>, <se:LineSymbolizer>, <se:PolygonSymbolizer>, <se:TextSymbolizer>, <se:RasterSymbolizer>	<SymbolSign>, <PolylineSign>, <SurfaceSign>, <TextSign>
Symbolizer opacity	<se:Opacity> inside <se:RasterSymbolizer> (no global opacity for vector symbolizers)	T parameter inside <Color> (no global opacity)
Selector	<fes:Filter>	WHERE clause with expression inside GRAPHIC in the graphic model
nestedRules	<fes:ElseFilter> provides limited support for one use case of nested rules	N/A

Tabelle 4: Styling Rules und Äquivalente zu SE und INTERLIS

Der Name einer "StylingRule" wird durch einen Ausdruck gefolgt von "Of" innerhalb des Elements "GRAPHIC" in der Standardsymbologievorlage beschrieben. Wenn die Regel nicht mit einem Selektor verknüpft ist, wird der Standardausdruck "Symbol" verwendet.

Wie im vorherigen Abschnitt erwähnt, definiert das INTERLIS-Standardsymbologiemodell die folgenden "Symbolizer": "PolylineSign", "SymbolSign", "SurfaceSign" und "TextSign". Mit anderen Worten kann man den OGC-Begriff "Symbolizer" und den INTER- LIS-Begriff "Sign(ature)" als gleichwertig betrachten.

Das INTERLIS-Symbologiemodell beschreibt die Opazität nicht global für eine "Symbolizer". Sie kann jedoch lokal im Zusammenhang mit einer Farbe angegeben werden.

Im INTERLIS-Symbolikmodell konnte kein Konzept identifiziert werden, das "nestedRules" ähnlich ist.

5.3.4. Symbolizers

Laut CartoSym beschreibt ein "Symbolizer", wie Geodaten (z. B. Vektoreinheiten oder Rasterdeckungsdaten) auf der Grundlage einer Reihe von Eigenschaften dargestellt werden können. Diese Anforderungsklasse definiert nur drei grundlegende Eigenschaften, die für die meisten Anwendungsfälle der Darstellung relevant sind. Separate Anforderungsklassen und spätere Teile erweitern diese Klasse um zusätzliche Eigenschaften, um fortgeschrittenere Darstellungsmöglichkeiten zu bieten.

Der Vergleich zwischen den Symbolisierern ("Symbolizers") und den Elementen des INTERLIS-Symbologiemodells wird in der folgenden Tabelle beschrieben.

CartoSym	SE	INTERLIS
visibility	<MaxScaleDenominator>, <MinScaleDenominator>	N/A
opacity	<Opacity> inside <RasterSymbolizer> (no global opacity for vector symbolizers)	T parameter inside <Color> (no global opacity)
zOrder	Order of <FeatureTypeStyles> in a <UserStyle> (sortBy and sortByGroup vendor options in GeoServer for additional control)	Priority

Tabelle 5: Symbolizers und Äquivalente SE und INTERLIS

5.3.5. Ausdrücke

Laut CartoSym sind "Ausdrücke" syntaktische Entitäten, die zur Bestimmung ihres Wertes ausgewertet werden können. Dieses Konzept hat eine ähnliche Bedeutung wie die in OGC Filter Encoding 2.0[67] definierten Ausdrücke und kann auch mit der Common Query Language (CQL2)[68] des OGC implementiert werden, vorausgesetzt, dass die hier definierten Ausdrücke nicht auf die Bewertung eines booleschen Wertes beschränkt sind, wie es bei der ersten Version von CQL2 der Fall ist.

Ausdrücke können gemäß dem konzeptuellen Modell Stile S Symbologie an zwei Stellen verwendet werden: als Selektor, der die Bedingung definiert, ob eine "Sty- lingRule" (Prädikat) angewendet wird oder nicht, sowie zur Festlegung der Werte von Eigenschaften des Symbolizers.

Die Ausdrücke: "Identifier Expressions", "Literal Expressions" und "Operation Expressions" werden in den folgenden Abschnitten definiert.

Für Instanzausdrücke ("Instance Expressions ") und Arrayausdrücke ("Array Expressions ") gibt es keine Entsprechungen, weder für SLD/SE noch für das INTERLIS-Symbologiemodell.

5.3.6. Parameterwerte

Nach CartoSym wird die Verwendung eines Ausdrucks als Eigenschaft des Symbolisierers als "Parameter Value". Ein Ausdruck kann ersetzt werden, wenn ein bestimmter Datentyp erwartet wird; in diesem Fall wird der Ausdruck entsprechend dem erwarteten Datentyp aufgelöst. Aus konzeptioneller Sicht können alle Eigenschaften des Symbolisierers mithilfe eines beliebigen Parameter-Value-Ausdrucks angegeben werden. Dies ist jedoch in dieser Basiskonformitätsklasse nicht vorgeschrieben, sondern wird durch spezifische Anforderungsklassen wie die Anforderungsklasse "Symbolizer Parameter Value Expressions" ermöglicht. In Ermangelung einer Anforderungsklasse, die das Gegenteil spezifiziert,

---

[67] https://www.ogc.org/standard/filter/

[68] https://www.ogc.org/standard/cql2/

nur literale Ausdrücke, die mit dem erwarteten Datentyp kompatibel sind, als Parameterwerte zugelassen werden.

Das Konzept der "Parameter Values" ist in der Vorlage StandardSymbology nicht möglich.

5.3.7. Ausdrücke identifizieren

Laut CartoSym wird ein " Identifier Expressions" durch eine Texteigenschaft repräsentiert, die zur Laufzeit aufgelöst wird.

Der Vergleich zwischen den Identifier Expressions ("Identifier Expressions") und den Elementen des INTERLIS-Symbologiemodells wird in der folgenden Tabelle beschrieben.

CartoSym	SE	INTERLIS
visualization	N/A	N/A
visualization.scaleDenominator	<se:MaxScaleDenominator>, <se:MinScaleDenominator>	N/A
visualization.dateTime	N/A	N/A
visualization.date	N/A	N/A
visualization.timeOfDay	N/A	N/A
zOrder & visualization.pass	Order of <se:FeatureTypeStyles> in a <sld:UserStyle> (sortBy and sortByGroup vendor options in GeoServer for additional control)	DrawingRule order & Priority in GRAPHIC in graphic model
feature.pass	Order of symbolizers in the <se:Rule>	N/A
dataLayer	N/A	N/A
dataLayer.identifier	<sld:NamedLayer> and <se:Name>	N/A
dataLayer.type	N/A	N/A

Tabelle 6: Identifizieren von Ausdrücken und Äquivalenten in SE und INTERLIS

Fast keine der Fähigkeiten der "Identifier Expressions" ist im INTERLIS-Symbologiemodell möglich, mit Ausnahme von "visualization.pass", das dem Attribut "Priori- ty" entspricht, das sich in einer "GRAPHIC" befindet, die im grafischen Modell definiert ist.

5.3.8. Literale Ausdrücke

Laut CartoSym ist ein "LiteralExpression" ein Ausdruck, der in der Stilcodierung einen festen Wert hat. Ein ähnliches Konzept einer literalen Klasse wurde ursprünglich in Sektio 7.5.1 des OGC-Standards Filter Encoding 2.0 definiert.

Der Vergleich zwischen literalen Ausdrücken ("Literal Expressions") und den Elementen des INTERLIS-Symbologiemodells wird in der folgenden Tabelle beschrieben.

CartoSym	SE	INTERLIS
TextLiteral	text directly in <fes:Literal>	TEXT in datamodel
BoolLiteral	true or false in <fes:Literal>	BOOLEAN type in DOMAIN in datamodel
NullLiteral	<fes:PropertyIsNull> when testing for null values	NULL in datamodel
IntegerLiteral	integer value directly in <fes:Literal>	1 .. 9999 in datamodel
RealLiteral	double value inside <fes:Literal>	NUMERIC in datamodel
ArrayLiteral	N/A	Domain in datamodel
InstanceLiteral	N/A	Domain in datamodel

Tabelle 7: Literale Ausdrücke und ihre Entsprechungen in SE und INTERLIS

Literale Ausdrücke ("Literal Expressions") werden alle auf Datentyp-Ebene im Datenmodell definiert.

5.3.9. Operation Expressions

Laut CartoSym führt ein "Operation Expression" eine von einem Operator spezifizierte Operation an Ausdrücken aus. Diese Klasse von Grundanforderungen definiert nur logische und relationale Operatoren. Die Implementierungen müssen auch Operationen zur Hierarchisierung unterstützen. Andere Anforderungsklassen führen die Unterstützung zusätzlicher Operatoren für arithmetische, textuelle und bi-nationale Operationen ein. Ohne die Unterstützung der Anforderungsklasse "Any right-hand operand" sind die Rechtsoperatoren von "RelationExpressions" auf "LiteralExpressions" beschränkt.

Der Vergleich zwischen den Operationsausdrücken ("Operation Expressions") und den Elementen des INTERLIS-Symbologiemodells wird in der folgenden Tabelle beschrieben.

CartoSym	SE	INTERLIS
RelationalOperator	Comparison operators (<fes:PropertyIsEqualTo>, <fes:PropertyIsLessThan>, etc.)	==, !=, <>, <=, >=, <, > with WHERE clause within GRAPHIC in graphic model
LogicOperator	Logical operators (<fes:And>, <fes:Or>, <fes:Not>)	AND, OR, NOT with WHERE clause within GRAPHIC in graphic model

Tabelle 8: Operation Expressions und Äquivalente SE und INTERLIS

Die relationalen Operatoren (RelationalOperator) entsprechen den Elementen: "==", "!=", "<>", "<=", ">=", "<", ">", die mit einer WHERE-Klausel im Element "GRAPHIC" des INTERLIS-Grafikmodells verknüpft werden können.

Logische Operatoren ("LogicOperator") entsprechen den Elementen "AND", "OR", "NOT", die mit einer WHERE-Klausel im Element "GRAPHIC" des INTERLIS-Grafikmodells verknüpft werden können.

5.3.10. Default class member values

Die Standardwerte für Klassenmitglieder werden nicht übernommen, da die Einstellungen bereits in den zuvor definierten Klassen vorkommen.

5.4. Anforderungsklasse "Basic Vector Features Styling".

Diese Anforderungsklasse fügt Unterstützung hinzu, um festzulegen, wie die Linie einer Vektorgeometrie gefüllt und gerendert werden soll. Darüber hinaus ermöglicht sie die Angabe eines Markers, der aus einer oder mehreren Grafiken besteht und an der exakten Position einer Punktgeometrie, am Zentroid einer polygonalen Geometrie oder an jedem Punkt einer linearen Geometrie gerendert werden soll.

Der Vergleich der Elemente der Anforderungsklasse "Basic Vector Features Styling" wird in den folgenden Abschnitten beschrieben.

5.4.1. Symbolizer extension for vector features

Die folgenden drei Eigenschaften des Symbolizers werden durch diese Anforderungsklasse definiert:

- Fill
- Stroke
- Marker

Der Vergleich zwischen der Symbolisierer-Erweiterung für Vektormerkmale ("Symbolizer extension for vector features") und den Elementen des INTERLIS-Symbologiemodells ist in der folgenden Tabelle dargestellt.

CartoSym	SE	INTERLIS
Fill	<se:Fill>	<ili:Name>fill</ili:Name> inside <SurfaceSign>
Stroke	<se:Stroke>	<Style> with reference to <LineStyle_> in <PolylineSign>
Marker	<se:PointSymbolizer>	<SymbolSign>

Tabelle 9: Symbolizer extension for vector features und Äquivalente zu SE und INTERLIS

Im Vergleich zur Erweiterung des Symbolizers[69] für Stile, die auf Vektordaten von CartoSym anwendbar sind, kann eine Füllung ("Fill") durch das Element <ili:Name>fill</ili:Name> innerhalb eines "SurfaceSign" in einer grafischen IN- TERLIS-Vorlage.

Im Vergleich zur Erweiterung des Symbolizers um Stile für CartoSym-Vektordaten kann eine Linie ("Stroke") mit dem Element "Style" in Bezug auf einen Linientyp ("LineStyle_") innerhalb eines "PolylineSign" beschrieben werden.

5.4.2. Extended Identifiers

Diese Anforderungsklasse definiert alle zugänglichen Eigenschaften eines abgebildeten Elements als gültige Identifikatoren, wobei der Name dieser Eigenschaften der Text des Identifikators ist. Diese Identifikatoren müssen mit dem Wert dieser Eigenschaft aufgelöst werden.

Der Vergleich zwischen den erweiterten Identifikatoren ("Extended Identifiers") und den Elementen des INTERLIS-Symbologiemodells wird in der folgenden Tabelle beschrieben.

CartoSym	SE	INTERLIS
feature.identifier	N/A	N/A
feature.geometryDimensions	N/A	N/A
feature.pass	Order of symbolizers in the <se:Rule>	Priority inside GRAPHIC in the graphic model
dataLayer.featuresGeometryDimension	N/A	N/A

Tabelle 10: Extended Identifiers und SE- und INTERLIS-Äquivalente

5.4.3. Farben

CartoSym definiert Farben mithilfe der Rot-, Grün- und Blaukomponenten. Die Möglichkeit, Farben in anderen Farbräumen (z. B. Lab, CMYK) zu definieren, wird in

[69] https://docs.ogc.org/DRAFTS/18-067r4.html#_symbolizer_extension_for_vector_features

eigene Anforderungsklassen und gilt überall dort, wo diese grundlegende RGB-Farbklasse verwendet wird.

Der Vergleich für die Farben ist in der folgenden Tabelle beschrieben.

CartoSym	SE	INTERLIS
Color	fill or stroke with <se:SvgParameter> (e.g., <se:SvgParameter name="fill">#000000</se:SvgParameter>)	Color

Tabelle 11: Farben und Äquivalente von SE und INTERLIS

Im INTERLIS-Symbologiemodell wird die Farbe nach dem Namensraum LCh mit dem zusätzlichen Parameter "T" für Transparenz definiert.

Auf der Ebene von CartoSym hingegen wird die Farbe nach den rgb-Einstellungen festgelegt.

Für die Interoperabilität von Kartenstilen empfehlen wir, mehrere Farbräume im Symbologiemodell zu definieren, sei es in CartoSym oder INTERLIS.

```
<Color ili:tid="1">
   <Name>white</Name>.
   <L>100.0</L>
   <C>0.0</C>
   <H>0.0</H>
   <T>1.0</T>
</Color>
```

Codeblock 11: Beispiel für die Definition einer Farbe (hier weiß)

5.4.4. Fills

Laut CartoSym definiert eine Füllung (oder "Fill") die grafischen Symbolisierungsparameter, die notwendig sind, um die Füllung einer zweidimensionalen Form, wie z. B. eines Polygons, zu zeichnen. Separate Konformitätsklassen können diese Klasse um zusätzliche Eigenschaften erweitern.

Der Vergleich für das Auffüllen ("Fills ") wird in der folgenden Tabelle beschrieben.

CartoSym	SE	INTERLIS
color	<se:SvgParameter name="fill"> for SE 1.1.0	<FillColor> with Color reference inside <SurfaceSign>
opacity	<se:SvgParameter name="fill-opacity"> for SE 1.1.0	<FillColor> with Color reference including a T parameter inside <SurfaceSign>

Tabelle 12: Fills und Äquivalente von SE und INTERLIS

Die Klasse SurfaceSign entspricht einem PolygonSymbolizer in SLD/SE. Die Füllfarbe wird durch einen Verweis ("FillColor") auf eine Farbe ("Color") innerhalb eines "SurfaceSign" in einer INTERLIS-Grafiksignatur beschrieben.

Die Transparenz ("opacity") wird durch den Parameter "T" der Farbe "Color" definiert, auf die sich die Füllung ("Fill") mit dem Parameter "FillColor" bezieht.

5.4.5. Strokes

Laut CartoSym definiert ein "Stroke" die Parameter der grafischen Symbolisierung, um eine Kontur zu des- inieren (z. B. bei linearen Geometrien oder der Außenseite einer polygonalen Geometrie).

CartoSym	SE	INTERLIS
color	<se:SvgParameter name="stroke"> for SE 1.1.0	Reference to <Color>in <FontSymbol_Polyline>
opacity	<se:SvgParameter name="stroke-opacity"> for SE 1.1.0	Reference to <Color>with T parameter in <FontSymbol_Polyline>
width	<se:SvgParameter name="stroke-width"> for SE 1.1.0	<Width> parameter inside <PolylineAttrs

Tabelle 13 : Stroke und Äquivalente in SE und INTERLIS

Die Klasse PolylineSign entspricht einem LineSymbolizer in SLD/SE und den Anforderungsklassen *8. Requirements Class "Basic Vector Features Styling*"[70] und *13. Requirements Classes for Advanced strokes*[71] in CartoSym.

Die folgende Signaturgrafik zeigt, wie man einen "Strokes" in IN- TERLIS darstellen kann.

```
<?xml version="1.0" encoding="UTF-8"?> <!-- File Strokes_Graphics_Signatures.xtf 2024-05-21 -->
```

[70] https://opengeospatial.github.io/ogcna-auto-review/18-067r4.html#toc23

[71] https://opengeospatial.github.io/ogcna-auto-review/18-067r4.html#toc38

```
<ili:transfer xmlns:ili="http://www.interlis.ch/xtf/2.4/INTERLIS"
   xmlns:geom="http://www.interlis.ch/geometry/1.0"
   xmlns="http://www.interlis.ch/xtf/2.4/StandardSymbology"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <ili:headersection>
      <ili:models>
         <ili:model>Polyline_Graphics</ili:model>
      </ili:models>
      <ili:sender>HEIG-VD</ili:sender>.
      <ili:comment>Basic example of Strokes Graphics Signatures</ili:comment>
   </ili:headersection>.
   <ili:datasection>
      <StandardSigns ili:bid="Strokes">.
         <!-- Color Library -->
         <Color ili:tid="1">
            <Name>schwarz</Name>
            <L>0.0</L>
            <C>0.0</C>
            <H>0.0</H>
            <T>1.0</T>
         </Color>
         <!-- Polyline Attribute -->
         <PolylineAttrs ili:tid="4001">
           <Width>0.01</Width>
         </PolylineAttrs>.
         <!-- Line Styles -->
         <LineStyle_Solid ili:tid="21">
            <Name>Continuous</Name>.
            <Color ili:ref="1"></Color>.
            <LineAttrs ili:ref="4001"></LineAttrs>.
         </LineStyle_Solid>.
         <!-- Polyline Signs -->
         <PolylineSign ili:tid="3001">
            <ili:Name>continuous</ili:Name>.
            <Style ili:ref="21">
              <PolylineSignLineStyleAssoc>.
              </PolylineSignLineStyleAssoc>.
            </Style>
         </PolylineSign>
      </StandardSigns>
   </ili:datasection>
</ili:transfer>
```

Codeblock 12: Beispiel[72] für die Definition eines Stroke in INTERLIS

[72] https://github.com/MediaComem/FGDM4GS/blob/main/WP2-WP3/model/symbology/Strokes.xtf

5.4.6. Markers

Um punktuelle Implantationen zu definieren, definiert INTERLIS die Klasse SymbolSign, die die Attribute Name, Scale, Rotation, Color, Symbol und ClipSymbol sowie die Attribute Geometry und Priority (obligatorisch) in der grafischen Vorlage enthält. Das Symbol wird, wie bereits erwähnt, in der Klasse FontSymbol beschrieben. Mit dem Parameter ClipSymbol kann eine Maske für das Symbol definiert werden. Die anderen Attribute wurden bereits in den anderen Grafiksignaturen (Sign) definiert.

OGC Symbology Encoding (SE) definiert das Konzept der Mark mit MarkIndex, um eine einzelne Glyphe aus einer Schriftdatei (Font) auszuwählen, was einem FontSym- bol entspricht (INTERLIS mit UCS4). SE erlaubt auch eine Mark, die durch einen WellK- nownName (Quadrat, Kreis, Dreieck, Stern, Kreuz und x) oder durch ein ExternalGraphic definiert ist, das eine externe Bitmap- oder Vektor-Bildressource (z. B. PNG, SVG) referenzieren kann.

CartoSym	SE	INTERLIS
marker	<PointSymbolizer>	<FontSymbol> with a <Geometry> or a reference to the glyph of a <Font> (<UCS4> parameter)
opacity	<SvgParameter name="fill-opacity"> for SE 1.1.0, <CssParameter name="fill-opacity"> for SLD 1.0.0 inside a <PointSymbolizer>	Reference to <Color>with T parameter in <SymbolSign>
position	<AnchorPoint> inside <PointSymbolizer>	<Geometry> inside <FontSymbol>
UnitPoint	uom attribute of symbolizers	px

Tabelle 14 : SE- und INTERLIS-Markers und -Äquivalente

In der StandardSymbology-Vorlage können Symbole entweder relativ zu einem Font (Glyphe) oder durch Beschreibung ihrer Geometrie in der Graphic Signature (Datei *.xtf). Die Verwendung einer externen Bildressource ("ExternalGraphic") ist hingegen nicht möglich.

In Anhang 8.2 wird eine grafische Signatur für einen Marker in INTERLIS exemplarisch dargestellt.

5.4.7. Dot Graphics

Laut CartoSym ist "Dot Graphic" ein durch diese Anforderungsklasse definierter Graphiktyp, der als Element eines Markers verwendet werden kann.

Der Vergleich von "Dot Graphics" wird in der folgenden Tabelle beschrieben.

CartoSym	SE	INTERLIS
dot	<WellKnownName> inside a <Mark> inside a <Graphic> inside a <PointSymbolizer>	<FontSymbol> with a <Geometry> or a reference to the glyph of a <Font> (<UCS4> parameter)
stroke	N/A	N/A

Tabelle 15: Dot Graphics und SE- und INTERLIS-Äquivalente

--

Das StandardSymbology-Modell bietet zwei Möglichkeiten, eine Dot-Grafik darzustellen. Die erste Möglichkeit besteht darin, die Geometrie des "Dot" über den Parameter "Geometry" innerhalb eines "FontSymbols" zu beschreiben. Dies ist jedoch angesichts der Anzahl der zu berücksichtigenden Koor- daten nicht ideal. Daher ist es besser, die Glyphe eines Fonts zu verwenden, indem Sie <FontSymbol> und eine "UCS4"-Referenz verwenden.

Die folgende Signaturgrafik zeigt, wie eine "Dot Graphics" in INTERLIS dargestellt werden kann.

```
<?xml version="1.0" encoding="UTF-8"?>.
<!-- File Dot_Graphics.xtf 2024-03-22 -->
<ili:transfer xmlns:ili="http://www.interlis.ch/xtf/2.4/INTERLIS"
  xmlns:geom="http://www.interlis.ch/geometry/1.0"
  xmlns="http://www.interlis.ch/xtf/2.4/StandardSymbology"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <ili:headersection>
      <ili:models>
         <ili:model>Point_Graphics</ili:model>
      </ili:models>
      <ili:sender>HEIG-VD</ili:sender>.
      <ili:comment>Basic example of Dot Graphics Signatures</ili:comment>
   </ili:headersection>.
   <ili:datasection>
      <StandardSigns ili:bid="Dot_Graphics">.
         <!-- Color Library -->
         <Color ili:tid="1">
            <Name>schwarz</Name>
            <L>0.0</L>
            <C>0.0</C>
            <H>0.0</H>
            <T>1.0</T>
         </Color>
         <!-- External Symbol Font "Symbols" -->
         <Font ili:tid="301">
            <Name>CadastraSymbol-Regular</Name>.
            <Internal>false</Internal>.
            <Type>Text</Type>
         </Font>
         <!-- FontSymbol Library-->
         <!-- Externe Symbole -->
         <!-- Tree -->
         <FontSymbol ili:tid="201">
            <Name>Dot</Name>
            <UCS4>0049</UCS4>
            <Font ili:ref="301"></Font>.
         </FontSymbol>
```

```
        <!-- Symbol Signs -->
        <SymbolSign ili:tid="2001">
           <ili:Name>Symbol</ili:Name>
           <Scale>1.0</Scale>
           <Color ili:ref="1"></Color>.
           <Symbol ili:ref="201"></Symbol>.
        </SymbolSign>
      </StandardSigns>
   </ili:datasection>
</ili:transfer>
```

Codeblock 13: Beispiel[73] für die Definition einer Mitgift in INTERLIS

5.4.8. Image Graphics

Laut CartoSym ist ein Bild ein von dieser Anforderungsklasse definierter Typ von Grafik, der als Teil eines Markers verwendet werden kann.

Der Vergleich von "Image Graphics" wird in der folgenden Tabelle beschrieben.

CartoSym	SE	INTERLIS
image	<Graphic>	<UCS4> & <Font> reference inside <FontSymbol>
hotSpot	<AnchorPoint> inside <PointSymbolizer>, or <AnchorPoint> inside <PointPlacement> inside <LabelPlacement> inside <TextSymbolizer>	N/A
tint	N/A	N/A
blackTint	N/A	N/A
alphaThreshold	N/A	N/A

Tabelle 16: Image Graphics und ihre Entsprechungen in SE und INTERLIS

Die einzige Möglichkeit, ein Bild in einer INTERLIS-Grafiksignatur darzustellen und auf ein Vektorbild zu verweisen, das zuvor in eine
"Font" und verwenden sie in einem "FontSymbol". Die Logik des Beispiels ist die gleiche wie bei " Dot Graphics ".

5.4.9. Textgrafiken

Laut CartoSym ist ein Text ein von dieser Anforderungsklasse definierter Grafiktyp, der als Teil einer Markierung verwendet werden kann.

[73] https://github.com/MediaComem/FGDM4GS/blob/main/WP2-WP3/model/symbology/Dot_Graphics.xtf

CartoSym	SE	INTERLIS
text	<WellKnownName> pointing to a TrueType Font with a <MarkIndex> identifiying the character inside the font inside a <Mark> inside a <Graphic> in a <PointSymbolizer> (only for single characters text)	<TextSign>
font	<Font> with <SvgParameter name="font-family">, <SvgParameter name="font-size">, <SvgParameter name="font-style">, <SvgParameter name="font-weight">	<Font> with <Name>, <Internal>, <Type> & <BottomBase>
alignment	<AnchorPoint> inside <PointSymbolizer>, or <AnchorPoint> inside <PointPlacement> inside <LabelPlacement> inside <TextSymbolizer>	<BottomBase> inside <Font>

Tabelle 17: Textgrafiken und ihre Entsprechungen in SE und INTERLIS

Erklärungen und ein Beispiel für eine grafische Signatur finden Sie im nächsten Abschnitt über die Kennzeichnung (Labeling).

5.5. Anforderungsklasse "Labeling".

5.5.1. Basic Labeling & Font Outlines

Diese Anforderungsklasse unterstützt Labels, die "über" geografischen Einheiten platziert werden, eventuell mithilfe eines Platzierungsalgorithmus. Wie Marker erben auch Labels von der Klasse MultiGraphic und können mehrere grafische Elemente enthalten.

Der Vergleich von "Basic Labeling S Font Outlines" wird in der folgenden Tabelle beschrieben.

CartoSym	SE	INTERLIS
label	<Label> inside <TextSymbolizer> (no support for image or shape labels in SLD 1.0.0)	<TextSign>
LabelPlacement	<LabelPlacement>	<BottomBase> inside <Font>
Font	<Font> with <SvgParameter name="font-family">, <SvgParameter name="font-size">, <SvgParameter name="font-style">, <SvgParameter name="font-weight">	<Font> with <Name>, <Internal>, <Type> & <BottomBase>
FontOutline	<Halo>	N/A

Tabelle 18: Basic Labeling und Font Outlines sowie SE- und INTERLIS-Äquivalente

Um Text-Implementierungen zu definieren, definiert INTERLIS die Klasse "TextSign", die die Attribute "Name", "Height", "Weight", "Slanted", "Underlined", "Stri- ked", "ClipBox", "Font" und "Color" sowie die Attribute "Geometry", "Rotation" und "TextSign" enthält.
"Priorität" (erforderlich) in der grafischen Vorlage. Die Parameter "Height", "Weight",
Mit den Einstellungen "Slanted", "Underlined" und "Striked" können Sie die Größe, das Gewicht, die Neigung, die Unterstreichung und das Durchstreichen des Textes festlegen. Mit dem Parameter "ClipBox" können Sie eine Begrenzungsbox für den Text festlegen. Die anderen Attribute wurden bereits in den anderen Grafiksignaturen ("Sign") definiert. Durch die Verwendung des Attributs "Spacing" wird der Abstand zwischen den einzelnen Elementen festgelegt.

in der Klasse "FontSymbol" endet, scheint nicht möglich zu sein, obwohl es mit Textsymbolen, wie sie in der Vorlage StandardSymbology definiert sind, dediziert zu sein scheint (siehe Bemerkung "*!! only for text symbols (characters)*"). In der Vorlage scheint eine Assoziation zu fehlen zwischen
"TextSign" und "FontSymbol".

Die untenstehende Grafiksignatur zeigt, wie ein "Basic Labeling" in INTERLIS dargestellt werden kann.

```
<?xml version="1.0" encoding="UTF-8"?>.
<!-- File Text_Graphics_Signatures.xtf 2024-03-22 -->
<ili:transfer xmlns:ili="http://www.interlis.ch/xtf/2.4/INTERLIS"
  xmlns:geom="http://www.interlis.ch/geometry/1.0"
  xmlns="http://www.interlis.ch/xtf/2.4/StandardSymbology"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <ili:headersection>
      <ili:models>
         <ili:model>Text_Grafiken</ili:model>
      </ili:models>
      <ili:sender>HEIG-VD</ili:sender>.
      <ili:comment>Basic example of Text_Graphics Signatures</ili:comment>
   </ili:headersection>.
   <ili:datasection>
      <StandardSigns ili:bid="Text_Graphics">.
         <!-- Color Library -->
         <Color ili:tid="1">
            <Name>schwarz</Name>
            <L>0.0</L>
            <C>0.0</C>
            <H>0.0</H>
            <T>1.0</T>
         </Color>
         <!-- External Text Font "Leroy" -->
         <Font ili:tid="11">
            <Name>Leroy</Name>.
            <Internal>false</Internal>.
            <Type>Text</Type>
            <BottomBase>0.3</BottomBase>
         </Font>
         <!-- Text Sign -->
         <TextSign ili:tid="1001">
            <ili:Name>Linefont_18</ili:Name>
            <Height>1.8</Height>
            <Weight>0.18</Weight>
            <Slanted>false</Slanted>
            <Underlined>false</Underlined>.
            <Striked>false</Striked>
```

```
            <ClipBox>.0</ClipBox>
            <Font ili:ref="11"></Font>.
            <Color ili:ref="1"></Color>.
         </TextSign>
      </StandardSigns>
   </ili:datasection>
</ili:transfer>
```

Codeblock 14: Beispiel[74] für die Definition eines Labels in INTERLIS

5.6. Anforderungsklasse "Advanced strokes".

Die Fähigkeiten der Anforderungsklasse "Advanced strokes" werden in den folgenden Abschnitten beschrieben.

5.6.1. Dashes

Diese Anforderungsklasse fügt Fähigkeiten für Striche hinzu, mit der Möglichkeit, das Muster von Strichen und Leerzeichen, die zum Malen des Strichs verwendet werden, sowie den Versatz auf der Ren- der zugehörigen Strich-Tabelle zu definieren.

Die folgende Tabelle zeigt den Vergleich der Dashes.

CartoSym	SE	INTERLIS
dashPattern	<CssParameter name="stroke-dasharray">	<DLength> inside <DashRec> inside <LineStyle_Dashed>
dashOffset	<CssParameter name="stroke-dashoffset">	<OffSet> in <PolylineSignLineStyleAssoc> in <PolylignSign>

Tabelle 19: Dashes und Äquivalente in SE und INTERLIS

Ein "DashPattern" kann durch das Attribut "DLength" der Struktur "DashRec" definiert werden, die durch die Klasse "LineStyle_Dashed" erweitert wird.

Ein "dashOffset" kann in einem "PolylineSign" durch den Parameter "Offset" der Assoziation "PolylineSignLineStyleAssoc" definiert werden.

Die untenstehende Grafiksignatur zeigt, wie man ein "dashPattern" und ein "dashOffset" in INTERLIS darstellt.

```
<?xml version="1.0" encoding="UTF-8"?>.
<!-- File Surface_Graphics_Signatures.xtf 2024-03-22 -->
<ili:transfer xmlns:ili="http://www.interlis.ch/xtf/2.4/INTERLIS"
   xmlns:geom="http://www.interlis.ch/geometry/1.0"
   xmlns="http://www.interlis.ch/xtf/2.4/StandardSymbology"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

[74] https://github.com/MediaComem/FGDM4GS/blob/main/WP2-WP3/model/symbology/Text_Graphics_Signatures.xtf

```
<ili:headersection>
   <ili:models>
      <ili:model>Polyline_Graphics</ili:model>
   </ili:models>
   <ili:sender>HEIG-VD</ili:sender>.
   <ili:comment>Basic example of Polyline_Graphics Signatures</ili:comment>
</ili:headersection>.
<ili:datasection>
   <StandardSigns ili:bid="Polyline_Graphics">.
      <!-- Color Library -->
      <Color ili:tid="1">
         <Name>schwarz</Name>
         <L>0.0</L>
         <C>0.0</C>
         <H>0.0</H>
         <T>1.0</T>
      </Color>
      <!-- Polyline Attribute -->
      <PolylineAttrs ili:tid="4001">
        <Width>0.01</Width>
        <!-- Join type -->
        <!-- Andere mögliche Werte sind: round, bevel -->.
        <Join>miter</Join>
        <!-- Hinweis: MiterLimit ist nur für den Typ miter join zulässig -->.
        <MiterLimit>10.0</MiterLimit>.
         <!-- Kaptyp -->
         <!-- Anderer möglicher Wert: rund -->
        <Caps>butt</Caps>
      </PolylineAttrs>.
      <!-- Line Styles -->
      <!-- Dashed -->
      <LineStyle_Dashed ili:tid="22">
         <Name>LineDashed_01</Name>
         <Dashes>
           <DashRec>
             <DLength>0.1</DLength>
           </DashRec>
         </Dashes>
         <Dashes>
           <DashRec>
             <DLength>0.1</DLength>
           </DashRec>
         </Dashes>
        <Color ili:ref="2"></Color>.
       </LineStyle_Dashed>.
```

```
        <!-- Polyline Signs -->
        <PolylineSign ili:tid="3001">
           <ili:Name>continuous</ili:Name>.
           <Style ili:ref="22">
             <PolylineSignLineStyleAssoc>.
               <Offset>0.0</Offset>
             </PolylineSignLineStyleAssoc>.
           </Style>
        </PolylineSign>
      </StandardSigns>
   </ili:datasection>
</ili:transfer>
```

Codeblock 15: Beispiel[75] für die Definition eines dashPattern und eines dashOffset in INTERLIS

5.6.2. "Casing und Centerline

Diese Anforderungsklasse definiert die Unterstützung von "Casing and Centerline".

Der Vergleich der Umschläge und Mittellinie ("Casing and Centerline") wird in der folgenden Tabelle beschrieben.

CartoSym	SE	INTERLIS
casing	multiple <FeatureTypeStyle> with different stroke widths	multiple <StandardSigns> with different stroke widths
center	multiple <FeatureTypeStyle> with different stroke widths	multiple <StandardSigns> with different stroke widths

Tabelle 20: Casing and Centerline und SE- und INTERLIS-Äquivalente

Ein "Casing" und ein "Center" können durch eine Überlagerung von "StandardSi- gns" definiert werden, die "LineStyle_Solid" oder "LineStyle_Dashed" enthalten.

Die untenstehende Signaturgrafik zeigt, wie man ein Casing und eine Center-Line in INTERLIS darstellen kann.

```
<?xml version="1.0" encoding="UTF-8"?> <!-- File Ply_Casing_Centerline_GS.xtf 2024-04-16 -->
<ili:transfer xmlns:ili="http://www.interlis.ch/xtf/2.4/INTERLIS"
  xmlns:geom="http://www.interlis.ch/geometry/1.0"    xmlns="http://www.interlis.ch/xtf/2.4/Stan-
dardSymbology" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <ili:headersection>
      <ili:models>
         <ili:model>Polyline_Graphics</ili:model>
      </ili:models>
      <ili:sender>HEIG-VD</ili:sender>.
      <ili:comment>Basic example of a Polyline Casing and Centerline Signatures</ili:comment>
   </ili:headersection>.
```

[75] https://github.com/MediaComem/FGDM4GS/blob/main/WP2-WP3/model/symbology/Surface_Graphics_Signatures.xtf

```
   <ili:datasection>
      <!-- 1st standard sign definition corresponding to the casing of a polyline -->.
      <StandardSigns ili:bid="Casing">.
         <Color ili:tid="1">
            <Name>schwarz</Name>
            <L>0.0</L>
            <C>0.0</C>
            <H>0.0</H>
            <T>1.0</T>
         </Color>
         <PolylineAttrs ili:tid="10">
            <Width>5</Width>
            <Join>miter</Join>
            <Caps>butt</Caps>
          </PolylineAttrs>.
          <LineStyle_Solid ili:tid="20">
             <Name>Casing</Name>
             <Color ili:ref="1"></Color>.
             <LineAttrs ili:ref="10"></LineAttrs>.
          </LineStyle_Solid>.
      </StandardSigns>
      <!-- 2nd standard sign definition corresponding to the centerline of a polyline -->.
      <StandardSigns ili:bid="Centerline">.
         <Color ili:tid="2">
            <Name>white</Name>.
            <L>100.0</L>
            <C>0.0</C>
            <H>0.0</H>
            <T>1.0</T>
         </Color>
         <PolylineAttrs ili:tid="11">
            <Width>3</Width>
            <Join>miter</Join>
            <Caps>butt</Caps>
         </PolylineAttrs>.
         <LineStyle_Solid ili:tid="21">
            <Name>Centerline</Name>.
            <Color ili:ref="2"></Color>.
            <LineAttrs ili:ref="11"></LineAttrs>.
         </LineStyle_Solid>.
      </StandardSigns>
   </ili:datasection>
</ili:transfer>
```

Codeblock 16: Beispiel[76] für die Definition eines Casings und Centers in INTERLIS

[76] https://github.com/MediaComem/FGDM4GS/blob/main/WP2-WP3/model/symbology/Ply_Casing_Centerline_GS.xtf

5.7. Anforderungsklasse " Advanced fills".

Die Fähigkeiten der Anforderungsklasse "Advanced fills" werden in den folgenden Abschnitten beschrieben.

5.7.1. "Hatch fills"

Diese Anforderungsklasse ermöglicht die Definition von Schraffurfüllungen in Form eines Musters aus Linien, die von :

- die Breite (Dicke) der Linie,
- der Abstand zwischen zwei Linien,
- den Winkel der Linien.

Der Vergleich der Schraffurfüllung wird in der folgenden Tabelle beschrieben .

CartoSym	SE	INTERLIS
HatchStyle	Extended (GeoServer) <WellknownName> prefixed by shape:// (vertline, horline, slash, backslash, plus, times) inside <Mark> inside <Graphic> inside <GraphicFill> and CssParameter such as stroke, stroke-width etc	<HatchSymb> with reference to <PolylineSign>in <SurfaceSign>

Tabelle 21: Hatch fills und Äquivalente SE und INTERLIS

Wie in der obigen Tabelle dargestellt, können mit dem Attribut "HatchSymb" der Assoziation "Surface- ceSignHatchSymbAssoc" "Hatch fills" dargestellt werden. Es muss auf ein "PolylineSign" in einer "SurfaceSign"-Klasse verweisen. Es ist anzumerken, dass diese Rückgabefähigkeit nicht mit SLD/SE existiert, sondern manchmal durch Softwareerweiterungen (VendorOption von GeoSer- ver) des Standards.

Die untenstehende Grafiksignatur stellt ein Beispiel für einen "HatchStyle" in INTERLIS dar.

```
<?xmlversion="1.0 "encoding="UTF-8"?>.
<!-- File Surface_Hatching_Stipples_GS.xtf 2024-04-17 -->
<ili:transferxmlns:ili="http://www.interlis.ch/xtf/2.4/INTERLIS"
 xmlns:geom="http://www.interlis.ch/geometry/1.0"
 xmlns="http://www.interlis.ch/xtf/2.4/StandardSymbology"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ili:headersection>
    <ili:models>
      <ili:model>Surface_Graphics</ili:model>
    </ili:models>
   <ili:sender>HEIG-VD</ili:sender>.
    <ili:comment>Basic example of Surface Hatching Signatures</ili:comment>
  </ili:headersection>.
  <ili:datasection>
    <StandardSignsili:bid="Surface_Graphics">.
      <!-- Color Library -->
      <Colorili:tid="1">
```

```
 <Name>white</Name>.
 <L>100.0</L>
 <C>0.0</C>
 <H>0.0</H>
 <T>1.0</T>
</Color>
<Colorili:tid="2">
 <Name>schwarz</Name>
 <L>0.0</L>
 <C>0.0</C>
 <H>0.0</H>
 <T>1.0</T>
</Color>
<!-- Polyline Attribute -->
<PolylineAttrsili:tid="4001">
 <Width>0.01</Width>
 <Join>round</Join>
 <Caps>butt</Caps>
</PolylineAttrs>.
<!-- Line Styles -->
<LineStyle_Solidili:tid="21">
 <Name>LineSolid_01</Name>.
 <Colorili:ref="2"></Color>.
 <LineAttrsili:ref="4001"></LineAttrs>.
</LineStyle_Solid>.
<!-- Surface Signs -->
<SurfaceSignili:tid="5001">
 <ili:Name>fill</ili:Name>.
 <FillColorili:ref="1"></FillColor>.
  <!-- Hatches -->
 <HatchOffset>2</HatchOffset>
  <!-- Hatch Symbol -->
 <HatchSymbili:ref="3001"></HatSymb>.
  <!-- Border Style definition -->
 <Borderili:ref="3001"></Border>.
</SurfaceSign>
<!-- Polyline Signs -->
<PolylineSignili:tid="3001">
 <ili:Name>continuous</ili:Name>.
 <Styleili:ref="21">.
  <PolylineSignLineStyleAssoc>.
    <Offset>0.0</Offset>
  </PolylineSignLineStyleAssoc>.
 </Style>
</PolylineSign>
```

```
    </StandardSigns>
  </ili:datasection>
</ili:transfer>
```

Codeblock 17: Beispiel[77] für die Definition eines HatchStyle in INTERLIS

5.7.2. "Dot Pattern fills"

Diese Anforderungsklasse erweitert die Füllfähigkeiten ("Fill") um die Unterstützung von Punktfüllungen mit einem auf Punkten basierenden Muster.

5.7.3. "Stipple fills"

Diese Anforderungsklasse definiert gepunktete Füllungen. Eine gepunktete Füllung ist ein Muster aus Punkten, das durch eine Dichte definiert ist.

Der Vergleich der gepunkteten Füllung ("StippleStyle") wird in der folgenden Tabelle beschrieben.

CartoSym	SE	INTERLIS
StippleStyle	Extended (GeoServer) `<WellknownName>` shape://dot inside `<Mark>` inside `<Graphic>` inside `<GraphicFill>` and CssParameter such as stroke, stroke-width etc	N/A

Tabelle 22: Stipple fills und Äquivalente SE und INTERLIS

Wie aus der obigen Tabelle hervorgeht, ist es mit dem aktuellen INTERLIS-Symbologiemodell nicht möglich, eine gepunktete Füllung ("StippleStyle") darzustellen.

Auch die StandardSymbology-Vorlage scheint das Ausfüllen mit einem "Pattern" (Muster) nicht zu unterstützen. Auch bleiben Fragen offen, wie etwa die Verwendung des Attributs Clip (inside, outside) der Vorlage StandardSymbology. Auch der Parameter HatchOrg scheint nicht richtig verwendet werden zu können.

5.8. Anforderungsklasse für zusätzliche Ausdrücke

5.8.1. Anforderungsklasse "Symbolizer Parameter Value Expressions".

Laut CartoSym ermöglicht diese Anforderungsklasse die Verwendung von ParameterValues, d. h. beliebigen Ausdrücken (nicht nur Literalen), für alle Eigenschaften des Symbolizers. Ein typisches Beispiel für die Verwendung dieses Konzepts ist die Darstellung in proportionalen Symbolen. Da dieses Konzept in der INTERLIS-Syntax nicht möglich ist, wird diese Anforderungsklasse nicht implementiert.

5.8.2. Anforderungsklasse "Any right-hand operands".

Laut CartoSym hebt diese Anforderungsklasse die Einschränkung auf, dass die Operanden auf der rechten Seite von RelationExpressions auf LiteralExpressions beschränkt sind. Für Ausdrücke, die in CQL2 codiert sind, ist sie analog zur CQL2-Anforderungsklasse für die

[77] https://github.com/MediaComem/FGDM4GS/blob/main/WP2-WP3/model/symbology/Surface_Hatching_Stipples_GS.xtf

Vergleich von Eigentum und Besitz. Die Verwendung von Ausdrücken ist in der INTERLIS-Syntax eingeschränkt und diese Anforderungsklasse ist nicht implementiert.

5.8.3. Anforderungsklasse "Conditional Expressions" (Bedingte Ausdrücke)

CartoSym definiert bedingte Ausdrücke ("condition", "thenExp" und "elseExp"), so dass eine "StylingRule" angewendet werden kann, wenn die definierten Kriterien erfüllt sind. In der INTERLIS-Syntax gibt es kein Analogon.

5.8.4. Anforderungsklasse "Variabel"

Laut CartoSym fügt diese Anforderungsklasse Unterstützung für die Definition von Variablen hinzu, die verwendet werden können, um die Wiederverwendung von Definitionen in Kodierungen zu erleichtern, sowie für die Bereitstellung von konfigurierbaren Elementen, die mit einer Anwendungssteuerung wie einem Schieberegler oder einem Dropdown-Steuerelement verbunden werden können. In der INTERLIS-Syntax gibt es kein Analogon.

5.8.5. Anforderungsklasse "Arithmetic Operators".

Laut CartoSym fügt diese Anforderungsklasse die Unterstützung für arithmetische Operationen hinzu. In der INTERLIS-Syntax gibt es kein Analogon.

5.8.6. Anforderungsklasse "Text Relation Operators".

Laut CartoSym fügt diese Anforderungsklasse eine fortschrittlichere Unterstützung für die Bewertung der Beziehung zwischen Textwerten hinzu, einschließlich des Operators "like", "starts with",
"contains", "ends with" sowie "not"-Varianten für alle diese Elemente. In der INTERLIS-Syntax gibt es keine Analoga.

5.8.7. Anforderungsklasse "Math Functions

Laut CartoSym fügt diese Anforderungsklasse Unterstützung für mathematische Funktionen hinzu, die jedoch nicht von INTERLIS unterstützt werden, was eine Erweiterung des internen Modells erforderlich machen könnte.

5.8.8. Anforderungsklasse "Array Relation Functions".

Laut CartoSym fügt diese Anforderungsklasse eine Unterstützung für die Handhabung von "Ar- rays" hinzu. In der INTERLIS-Syntax existiert kein Analogon, was eine Erweiterung des Modells erforderlich machen könnte.

5.8.9. Anforderungsklasse "Text Manipulation Functions".

Laut CartoSym fügt diese Anforderungsklasse eine Unterstützung für die Textmanipulation hinzu. In der INTERLIS-Syntax existiert kein Analogon, was eine Erweiterung des Modells erforderlich machen könnte.

5.9. Beispiele auf der Grundlage der ausgewählten Vorlagen

5.9.1. Beispiel 1: Achsen von Nationalstraßen

Das 1er Beispiel wird auf der Grundlage des MGDM National Road Axes (86.1) definiert. Das Darstellungsmodell und die zugehörigen grafischen Signaturen werden auf der Grundlage der VIEW in Abschnitt 4: Nationalstraßenachsen definiert, der das Attribut "Typ" ausdrücklich hinzugefügt wurde, um die grafische Darstellung zu verwenden. Gemäß der Dokumentation der Vorlage[78] muss die Grafik auf der Grundlage des Wertes des Attributs "AxisType" definiert werden, das in einem Objektkatalog enthalten ist, der über das Vorlagenrepositorium des Bundes zugänglich ist[79] .

Die Darstellung des MGDM[80] mobilisiert 4 verschiedene Objekte:

- Das Objekt "Achsensegment", das einem "PolylineSign" entspricht
- Das Objekt "Sektor" (Orientierungspunkt), das einem "SymbolSign" entspricht
- Das Objekt "Kalibrierungspunkt", das einem "SymbolSign" entspricht
- Das Objekt "physische Materialisierung", das einem "SymbolSign" entspricht

Abbildung 12: Kartendarstellung des MGDM Nationalstraßen-Achsen

Probleme:

1. Die Definition des grafischen Modells für das Objekt "Achsensegment" ist nicht möglich, da sie auf den Werten der TIDs der Klasse "AxisCatalogs_V1_1.AxisCatalogs.AxisType" des oben genannten Katalogs AxisCatalogs_V1_1 basiert, die von der "DrawingRule" nicht unterstützt werden (Fehler: *.\AxisAxis_V1_1_gm.ili:15:expecting NAME, found '89'*). Dieser Fehler bezieht sich jedoch nur auf den Objektkatalog, der aktualisiert werden könnte, um die Kriterien der DrawingRule zu erfüllen.

[78] https://www.astra.admin.ch/astra/fr/home/services/weitere-bereiche/geoinformation/geodonnees-de-base/reseau-des-routes-nationale.html

[79] https://models.geo.admin.ch/ASTRA/AxisCatalogs_V1_1_20200205.xml

[80] https://www.astra.admin.ch/dam/astra/fr/dokumente/Geoinformation/beschreibung_desminimalengeodatenmodells-national-strassennetz.pdf.download.pdf/beschreibung_des_Modells_zur_Minimalgeodatenerfassung_des_Nationalstraßennetzes.pdf

2. Die Definition des grafischen Modells für das Objekt "Sektor" (Landmarke) ist ebenfalls nicht möglich, da sie den Rückgriff auf die Werte des Attributs "Sequence" der Klasse "Sector" erfordert, was in INTERLIS derzeit nicht möglich ist (siehe Abschnitt Bemerkungen zur Anforderungsklasse Symbolizer Parameter Value Expressions).

3. Die Definition von grafischen Modellen für die Objekte "Kalibrierungspunkt" und "physische Mate- rialisierung" ist nicht möglich, da es in der Modelldokumentation keine Details gibt, die es ermöglichen, mit Sicherheit zu bestimmen, auf welche Klassen und Zuordnungen sie sich beziehen.

Die Grundlage für die Definition der grafischen Vorlage finden Sie weiter unten.

```
INTERLIS 2.3;
!! Version |            | Who     | Bearbeiten
Datum
!!------------------------------------------------------------------------------------
!!1.O      2024-01-04 | HEIG-VD | - Erstellen der
Vorlage MODEL Axis_V1_1_gm
AT "mailto:maxime.collombin@heig-vd.ch"
  VERSION "2024-01-04" =
  IMPORTE Axis_LV95_V1_1_d;
  !! Der Import der Vorlage Axis_LV95_V1_1 ist auch gegenüber der Klasse Sector IMPORTS
  Axis_LV95_V1_1 erforderlich;
  IMPORTIERT StandardSymbology;
  !! SIGN BASKET StandardSymbology ~ StandardSymbology.StandardSigns
    !! !! Definition der Objekte "Achsensegment" (PolylineSign) auf der Grundlage der Klasse
AxisCat- alogs_V1_1.AxisCatalogs.AxisType aus dem Katalog AxisCatalogs_V1_1
    !! OBJECTS OF PolylineSign: 89aa138a-999a-554e-b120-13721dbbb055, 1f22835b-e726-bd40-b95d-
5b5de18db20a, bf44d30a-b831-c242-beb3-1b252d7be688 ;
    !! !! TIDs werden von der DrawingRule nicht unterstützt (Fehler:
.\Axis_V1_1_gm.ili:15:expect- ing NAME, found '89')
    !! !! Definition von Objekten "Sektor" (Orientierungspunkt) (SymbolSign)
    !! OBJECTS OF SymbolSign: ;
    !! !! Die Definition dieser DrawingRule ist nicht möglich, da sie die Verwendung von
ParameterValues erfordert
  !! TOPIC-Definition für GRAPHICs ( ~ styleSheets) TOPIC
  Graphics =
    DEPENDS ON Axis_LV95_V1_1_d.Axis_d;
    DEPENDS ON Axis_LV95_V1_1.Axis;

    !! GRAPHIC (styleSheet) für Achsensegmente definieren
    GRAPHIC AxeSegment
      BASED ON Axis_LV95_V1_1_d.Axis_d.view_Axis =
      !! Definitionen von DrawingRules (stylingRules)
    END AxeSegment;¶

    !! GRAPHIC (styleSheet) für Orientierungspunkte festlegen
```

```
    !! Unmögliche Definitionen aufgrund von nicht unterstützten TIDs
    GRAPHIC Sector
      BASED ON Axis_LV95_V1_1.Axis.Sector =
      !! Definitionen von DrawingRules (StylingRules)
      !! Definitionen nicht möglich, da Zugriff auf die Werte des Attributs Sequence der Klasse
Sector (ParameterValue) erforderlich ist.
    END Sector;¶

    !! GRAPHIC (styleSheet)-Definition "Kalibrierungspunkt" (nicht möglich, da keine
Informationen verfügbar sind)
    !! GRAPHIC Kalibrierungspunkt
    !!   BASED ON <Klasse & Attribut nicht spezifiziert> =
    !! END Kalibrierungspunkt;
    !! Definition von GRAPHIC (styleSheet) "physische Materialisierung" (mangels verfügbarer
Informationen nicht möglich)
    !! GRAPHIC PhysikalischeVersicherung
    !!   BASED ON <Klasse & Attribut nicht spezifiziert> =
    !! END PhysikalischeVersicherung;
  END Graphics;
END Axis_V1_1_gm.
```

Codeblock 18: Grafisches Modell[81] *des MGDM Nationalstraßen-Achsen*

5.9.2. Beispiel 2: Inventar historischer Verkehrswege

Das 2ème Beispiel wird auf der Grundlage des MGDM Bundesinventar historischer Verkehrswege (16.1, 17.1) definiert. Das Darstellungsmodell und die damit verbundenen grafischen Signaturen werden auf der Grundlage der VIEW des Beispiels 2 in Abschnitt 4: Inventar historischer Verkehrswege definiert. Die Darstellung des MGDM[82] mobilisiert 2 Arten von Implantaten: linear (PolylineSign) basierend auf der Klasse "ivs_linienobjekte_lv95" und punktuell (SymbolSign) basierend auf der Klasse "ivs_punktobjekte_lv95".

Probleme :

- Der Vergleich von literalen Werten erfordert in der Regel eine Zeichenkette, bei der Anfang und Ende klar definiert sind (z. B. Anfang und Ende mit Anführungszeichen definiert). Die vorhandenen INTERLIS-Werkzeuge können mit einer Zeichenkette, die aus Wörtern mit Leerzeichen besteht, nur schlecht umgehen.
- SymbolSign erfordert die Verwendung der Glyphen des Fonts *ivs200440903*, wie in der Dokumentation der Vorlage erwähnt. Diese Schriftart ist jedoch nirgends zugänglich.

---

[81] https://github.com/MediaComem/FGDM4GS/blob/main/WP2-WP3/model/Axis_V1_1/Axis_V1_1_gm.ili

[82] https://www.astra.admin.ch/dam/astra/fr/dokumente/Geoinformation/beschreibung_minimalesgeodatenmodells-historischev-erkehrswege.pdf.download.pdf/beschreibung_des_geodätischen_Modellshistorischer_Kommunikationswege.pdf

5.9.3. Beispiel 3: Reservierte Bereiche

Das dritte Beispiel wird auf der Grundlage des MGDM Reservierte Zone (76.1) definiert. Das Darstellungsmodell und die zugehörigen grafischen Signaturen werden auf der Grundlage der VIEW des Beispiels 3bis in Abschnitt 4: Reservierte Gebiete definiert. Die "DrawingRule" der grafischen Vorlage wird anhand des Attributs: "festlegung_stufe_typ" definiert. Die folgenden Codeauszüge beziehen sich auf die Vorlage und die Definition der grafischen Signaturen.

```
INTERLIS 2.4;V1.O        2023-12-22 | HEIG-VD | - Erstellung des Modells
MODEL Planungszonen_V2_gm
AT "mailto:maxime.collombin@heig-vd.ch"
  VERSION "2023-12-22" =
  IMPORTS Planungszonen_V2_d_B;
  IMPORTS StandardSymbology;
  SIGN BASKET StandardSymbology ~ StandardSymbology.StandardSigns
    OBJECTS OF SurfaceSign: Kanton, Gemeinde, andere;
  TOPIC Grafiken =
    DEPENDS ON Planungszonen_V2_d_B.PZ_d;
    GRAPHIC Surface_Graphics
      BASED ON Planungszonen_V2_d_B.PZ_d.view_pz =
      Kanton OF StandardSymbology.StandardSigns.SurfaceSign:
        WHERE festlegung_stufe_typ == #Kanton (
          Sign := {Kanton};
          Geometry := wkb_geometry;
          Priority := 100);
      Gemeinde OF StandardSymbology.StandardSigns.SurfaceSign:
        WHERE festlegung_stufe_typ == #Gemeinde (
          Sign := {Gemeinde};
          Geometry := wkb_geometry;
          Priority := 100);
      andere OF StandardSymbology.StandardSigns.SurfaceSign:
        WHERE festlegung_stufe_typ == #andere (
          Sign := {andere};
          Geometry := wkb_geometry;
          Priority := 100);
    END Surface_Graphics;
  END Graphics;
END Planungszonen_V2_gm.
```

Codeblock 19: Grafisches Modell[83] des MGDM Reservierte Gebiete

[83] https://github.com/MediaComem/FGDM4GS/blob/main/WP2-WP3/model/Planungszonen_V1_1/Planungszonen_V2_gm.ili

```
<?xml version="1.0" encoding="UTF-8"?> <!-- File Planungszonen_V2_Symbols.xtf 2024-01-03 -->
<ili:transfer xmlns:ili="http://www.interlis.ch/xtf/2.4/INTERLIS"
  xmlns:geom="http://www.interlis.ch/geometry/1.0"
  xmlns="http://www.interlis.ch/xtf/2.4/StandardSymbology"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:Planungszonen_V1_1_gm="http://www.interlis.ch/xtf/2.4/Planungszonen_V1_1_gm">
  <ili:headersection>
    <ili:models>
      <ili:model>StandardSymbology</ili:model>
      <ili:model>Planungszonen_V2_gm</ili:model>
    </ili:models>
    <ili:sender>HEIG-VD</ili:sender>.
    <ili:comment>Planungszonen_V2 Grafic Modell Signatures</ili:comment>
  </ili:headersection>.
  <ili:datasection>
    <!-- Definiere die StandardSigns -->.
    <StandardSigns ili:bid="PlanungszonenSymb">.
        <!-- Farbdefinition -->
        <Color ili:tid="KantonFarbe">.
            <Name>C-Violett_mittel</Name>
            <L>81.69</L>
            <C>34.96</C>
            <H>308.4</H>
            <T>0.6</T>
        </Color>
        <Color ili:tid="GemeindeFarbe">
            <Name>C-Hellviolett</Name>
            <L>88.57</L>
            <C>21.09</C>
            <H>306.48</H>
            <T>0.6</T>
        </Color>
        <Color ili:tid="AndereFarbe">.
            <Name>C-Purpur</Name>.
            <L>94.86</L>
            <C>8.59</C>
            <H>297.89</H>
            <T>0.6</T>
        </Color>
        <Color ili:tid="BorderFarbe">.
            <Name>C-Dunkelviolett</Name>
            <L>73.7</L>
            <C>37.41</C>
            <H>301.72</H>
            <T>1</T>
```

```
        </Color>
        <!-- Definiere die Fills -->.
        <SurfaceSign ili:tid="KantonFill">.
            <ili:Name>Kanton</ili:Name>.
            <FillColor ili:ref="KantonFarbe"></FillColor>.
          </SurfaceSign>
          <SurfaceSign ili:tid="GemeindeFill">.
            <ili:Name>Gemeinde</ili:Name>
            <FillColor ili:ref="GemeindeFarbe"></FillColor>.
          </SurfaceSign>
          <SurfaceSign ili:tid="AndereFill">.
            <ili:Name>Andere</ili:Name>.
            <FillColor ili:ref="AndereFarbe"></FillColor>.
          </SurfaceSign>
          <!-- Definiere die BorderWidth -->.
          <PolylineAttrs ili:tid="BorderWidth">
              <Width>2</Width>
          </PolylineAttrs>.
           <!-- Definiere den BorderStyle -->.
           <LineStyle_Solid ili:tid="BorderStyle">.
              <Name>BorderStyle</Name>
              <Color ili:ref="KantonFarbe"></Color>.
              <LineAttrs ili:ref="BorderWidth"></LineAttrs>.
           </LineStyle_Solid>.
          <!-- Definiere die Grenze -->.
          <PolylineSign ili:tid="Border">.
              <ili:Name>Border</ili:Name>
              <Style ili:ref="BorderStyle">.
                <PolylineSignLineStyleAssoc>.
                  <Offset>0.0</Offset>
                </PolylineSignLineStyleAssoc>.
              </Style>
          </PolylineSign>
    </StandardSigns>
  </ili:datasection>
</ili:transfer>
```

Codeblock 20: Grafische Signaturen[84] des MGDM Reservierte Gebiete

5.9.4. Beispiel 4: Sachplan Verkehr Teil Infrastruktur Straße

Das 4ème Beispiel wird auf der Grundlage des MGDM Sektorplan Verkehr Teil Infrastruktur Straße (72.1) definiert. Das Darstellungsmodell und die zugehörigen grafischen Signaturen werden auf der Grundlage der VIEW des Beispiels 4 in Abschnitt 4: Verkehrswegeplan definiert.

---

[84] https://github.com/MediaComem/FGDM4GS/blob/main/WP2-WP3/model/Planungszonen_V1_1/Planungszonen_V2_Symbols.xtf

toriel des transports Teil Infrastruktur Straße. Gemäß der Dokumentation des Modells[85] müssen die Grafiken auf der Grundlage der Werte der Attribute "FacilityStatus" und "FacilityStatus" definiert werden.
"PlanningStatus" Referenten in einem Objektkatalog, der über den Vorlagenspeicher der Konföderation zugänglich ist .[86]

Problem: Da die Werte des Attributs "FacilityKind" der Klasse "Facility" des Modells "BaseModel_SectoralPlans_Catalogues_V1_4" (auf Ebene des TOPIC "SectoralPlans_WithLatestModification") nicht im Katalog "SectoralPlans_Catalogues_V1_4" definiert sind, ist es nicht möglich, das Modell und die grafischen Signaturen zu definieren. Dies ist jedoch kein Problem des StandardSymbology-Modells, sondern ein Problem der MGDM-Implementierung.

5.10. Empfehlungen

5.10.1. Rasterdarstellung

Die Vorlage StandardSymbology könnte aktualisiert werden, um eine Rasterdarstellung zu ermöglichen ("RasterSymbolizer" eines "CoverageStyle" für OGC SE oder Anforderungsklasse "Co- verage Styling" für CartoSym).

5.10.2. Farbräume

Für die Interoperabilität von Kartenstilen empfehlen wir, mehrere Farbräume im Symbologiemodell zu definieren, sei es in CartoSym oder INTERLIS.

5.10.3. Formatierung von Literalausdrücken

Für die Selektoren in der grafischen Vorlage sind die Ausdrücke eingeschränkt. Sie können nur grundlegenden numerischen Werten oder Strings entsprechen, ohne Leerzeichen, Akzente oder Sonderzeichen (siehe z. B. die Vorlagen in der Datei IVS_V3_gm.ili [87] `(WHERE ivs_signatur_label == #Nationale Bedeutung, historischer Verlauf mit viel Substanz`). Es wäre wichtig, in eCH-0031 die Formatierung von wörtlichen Ausdrücken zu klären (z.B. `WHERE myAttribute == 'my space delimited value'`) .

5.10.4. Transkodierer für Stil/Symbologie

Um die Interoperabilität zwischen INTERLIS und anderen Beschreibungssprachen für Kartenstile zu fördern, sollte zunächst ein Mapping zwischen INTERLIS, OGC SE S CartoSym erstellt und dann *Parser/Writer* implementiert werden, um die IN- TERLIS-Definition beispielsweise in CartoSymJSON zu transkribieren.

---

[85] https://www.astra.admin.ch/astra/fr/home/services/weitere-bereiche/geoinformation/geodonnees-de-base/plan-sectoriel-des-verkehrs-teil-strasseninfrastruktur.html

[86] https://models.geo.admin.ch/ARE/SectoralPlans_Catalogues_V1_4.xml

[87] https://github.com/MediaComem/FGDM4GS/blob/main/WP2-WP3/model/IVS_V2_1/IVS_V3_gm.ili

5.10.5. Erweiterung des INTERLIS-Modells zur Beschreibung von Symbologien

Die folgenden Anforderungsklassen sind nicht oder nur teilweise in das INTERLIS-Modell zur Beschreibung von Symbologien integriert

- Symbolizers (Sichtbarkeit)
- Ausdrücke identifizieren
- Image GraphicsStipple fills
- Anforderungsklasse für zusätzliche Ausdrücke
- Anforderungsklasse für Funktionen

Einige Anforderungen könnten StandardSymbology bereichern, andere könnten z. B. Gegenstand eines ergänzenden Modells AdvancedSymbology sein.

5.10.6. Integration von AbstractSymbology und StandardSymbology

Das AbstractSymbology-Modell könnte potenziell in das Stan- dardSymbology-Modell integriert werden (wie übrigens auch von M. Baertschi in seiner Bache- lorarbeit an der ETHZ empfohlen).[88]

5.10.7. Bereitstellung von Fonts in einem Repository

Wenn die grafische Darstellung spezielle Ressourcen für eine Grafik verwendet, ist es gut, wenn die Fonts online zugänglich sind (z. B. WESP Font) und die Exter- nalGraphic ebenfalls über eine GetStyles-Operation verfügbar sind, oder aber es sollten Ressourcen vom Typ SVG bevorzugt werden.

5.10.8. Stil-Katalog

Es wäre interessant, Sammlungen von Stilen zu veröffentlichen, um ihre Wiederverwendung und die Harmonisierung der kartografischen Darstellungen der veröffentlichten Geodienste zu fördern. Ein Beispiel hierfür wäre die OGC API - Styles Spezifikation.

Abbildung 13: Konzept für die Verknüpfung von NGDI-Ressourcen und die Umsetzung einer Stil-Kolumne

[88] M. Bärtschi (2007) INTERLIS 2 Grafikmodellierung und Anforderungen der Kartografie. Bachelorarbeit, Geomatik und Planung ETHZ.

5.10.9. Normativer Ansatz für die Symbologie

Die Bereitstellung von Stilen mit Hilfe von INTERLIS kann mehrere Vorteile haben - wie z.B.: einheitliche Definition von Stilen; Bereitstellung von Stilen am gleichen Ort; Möglichkeit, die Erstellung von Stilen in verschiedenen Formaten auf der gleichen Basis zu automatisieren. Es wird vorgeschlagen, einen normativen Ansatz zu wählen und die Empfehlungen von 2014 z.B. durch einen neuen eCH-Standard zu ersetzen und auch die Standards eCH-0031 und eCH- 0056 anzupassen.

6. Automatisierung der Erstellung von Geodiensten

In Anbetracht der Beschränkungen, die die Empfehlungen aufheben sollen (insbesondere des ili2db-Tools), kann eine Automatisierung nicht implementiert werden. Wenn man sie jedoch als le- viert ansieht, dann würde ein solcher Prozess aus den folgenden Schritten bestehen:

- Schritt 1: Wie in Kapitel 4 gezeigt und gemäß den Empfehlungen, geht es darum, eine Ansicht durch die Definition eines abgeleiteten Modells MGDM zu erstellen, die der Veröffentlichung von Geodiensten gewidmet ist.
- Schritt 2a: Verwendete das Werkzeug ili2db, um die Definitionen des IN- TERLIS-Datenmodells in ein Datenbankschema (PostgreSQL/PostGIS, GeoPackage, ESRI FileGDB) zu übersetzen und die INTERLIS-Daten in die erstellte Datenbank zu laden.

  nb. erfordert Unterstützung JOIN OF ... WHERE (gemäß Empfehlungen), um die entsprechende Ansicht im physischen Modell der Zieldatenbank zu erstellen.
- Schritt 2b: Transkodieren Sie die Definition des Darstellungsmodells MGDM in die von den normativen Grundlagen vorgeschriebene Form der Stilbeschreibung/Symbolik.

  nb. erfordert ein Transcodierungswerkzeug (gemäß Empfehlungen), dessen Konformität mit den OGC-Standards für die Modellierung von Stilen/Symbologien die Interoperabilität von Geodiensten fördert (z. B. OGC SymCore/CartoSym, OGC API Styles).
- Schritt 3: Erstellen und Verteilen der Konfigurationsdateien der Geodienste in der Veröffentlichungsinfrastruktur (z.B. für mapserver, arcgis server, qgis-server, etc.)

  nb. nach normativen Grundlagen wie eCH-0056 für Dienste zum Herunterladen von Daten (z. B. rund um die OGC API Features), für Dienste zum Abrufen von Daten (z. B. heute rund um Basic WMS und Styleable WMS, morgen ähnliche Anforderungen rund um die OGC API Maps und Styles, usw.).

7. Schlussfolgerung

Das Projekt FGDM4GS hat die Möglichkeiten und Grenzen der Verwendung von VIEW und der Modellierung von Darstellungen mit grafischen Signaturen aufgezeigt, um die Veröffentlichung von denormalisierten Geodatensätzen und Geodiensten zu fördern, während das INTERLIS-Basismodell unberührt bleibt.

Auf der Grundlage von vier ausgewählten INTERLIS-Modellen wurden die abgeleiteten Modelle und die Darstellungsmodelle so weit wie möglich implementiert. Dennoch kann man einen Mangel an Informationen und Dokumentationen feststellen. Die Kompilierungs- und Validierungswerkzeuge waren nicht immer hilfreich, und trotz zahlreicher Kontakte mit den Akteuren der INTERLIS-Gemeinschaft, um diese Schwierigkeiten zu überwinden, sind die erarbeiteten Beispiele weiterhin bedenkenswert. In gewisser Weise bestätigt dies eine Art "Unreife" der INTERLIS-Umgebung und die Grenzen der zugrunde liegenden Praktiken für die Symbologie. Dies gilt auch für die in Abschnitt 5.10 formulierten Empfehlungen.

Auch wenn es derzeit noch einige Einschränkungen gibt, wie sie in den Re- Commandations beschrieben werden, könnte es für die Umsetzung der MDGs von großem Interesse sein, sie zu berücksichtigen.

Es wird sinnvoll sein, im Anschluss an dieses Projekt einen Workshop zu organisieren, um die nützlichen Überlegungen mit Referenten aus der INTERLIS-Gemeinschaft fortzusetzen und einen Aktionsplan festzulegen. Um das volle Potenzial zu entfalten und der Problematik dieses Projekts gerecht zu werden, haben wir die folgenden Empfehlungen formuliert (siehe Kapitel 4.4 und 5.10):

1. Erstellen einer abgeleiteten Vorlage
2. Ansicht vom Typ Projektion (PROJECTION OF)
3. Ansicht vom Typ Verbindung (JOIN OF)
4. ili2db-Unterstützung vom Typ Projektion und Verbindung
5. Normativer Ansatz für Ansichten
6. Rasterdarstellung
7. Farbräume
8. Formatieren von literalen Ausdrücken
9. Transkodierer für Stil/Symbologie
10. Erweiterung des INTERLIS-Modells zur Beschreibung von Symbologien
11. Integration von AbstractSymbology und StandardSymbology
12. Bereitstellung von Fonts in einem Repository
13. Stil-Katalog
14. Normativer Ansatz für die Symbologie

Im Hinblick auf einen Aktionsplan scheinen sich aus diesen Empfehlungen zwei Dimensionen zu ergeben:

- normative Dimension: Wir schlagen sowohl einen normativen Ansatz für die Erstellung von INTERLIS VIEW als auch einen normativen Ansatz für die Erweiterung der bestehenden INTERLIS-Symbologie vor: Erneuerung der Empfehlungen zu Darstellungsmodellen von 2014 und die Entwicklung eines neuen eCH-Standards sowie parallel dazu eine Anpassung der Standards eCH-0031 und eCH-0056.
- Werkzeugdimension ("tooling"): Wir schlagen eine Anpassung der INTER- LIS-Werkzeuge ili2c (Compiler) und ili2db vor, um die INTERLIS-VIEWs (vorrangig JOIN OF und PROJECTION OF) zu berücksichtigen. Außerdem soll die Möglichkeit der Transkodierung von INTER- LIS-Stilen in die geltenden Normen und Interoperabilitätsstandards in Betracht gezogen werden.

8. Anhänge

8.1. Verfahren zur Auswahl der für diese Studie gegebenen Modelle

Description of the performed tasks :

- Erstellung eines Python-Skripts, um Daten aus der geobasisdaten.ch API in eine Excel-Datei zu extrahieren (um nur die benötigten Spalten zu behalten).
- Entfernen von Einträgen, die kein .ili-Datenmodell haben
- Manuelle Suche nach den relevanten Websites für Links zu den Darstellungsmodellen (pdf, zip S excel)
- Verwendung des Linux-Skripts: "check_xlsx.sh", um zu testen, ob Zips Excel-Dateien enthalten.
- Löschen von Modelleinträgen, die nicht über Excel-Dateien zur Darstellung von Modellen verfügen. Wir gehen davon aus, dass diese excel-Dateien das standardisierte Beschreibungsformat repräsentieren und ig- nore die anderen.
- Verwendung des Linux-Skripts "getfilename.sh", um die Namen von Excel-Dateien auszudrucken (bei Zip-Dateien, die mehrere Dateien enthalten, werden die Namen dieser Dateien zu einer Zeichenkette verkettet, die in eine Excel-Zelle übertragen wird).
- Verwendung von Excel-Funktionen (=A2=A1; =AND(A1=TRUE,B1=TRUE)), um zu bestimmen, welche Modelle (Daten und Darstellung) mehrfach aufgelistet werden, dann Aggregation der zugehörigen Informationen und Löschung überflüssiger Einträge.
- Um die zu verwendenden Modelle auszuwählen, haben wir folgende Auswahlkriterien definiert: - Einträge in geobasisdaten.ch, die einen Link zu einem Datenmodell (.ili-Datei) haben - Einträge aus geosbasisdaten.ch, die einen Link zu einem .xlsx-Darstellungsmodell in Excel for- mat (weil diese Dateien den standardisierten Ressourcen am nächsten kommen und die Darstellungsmodelle mit der Definition von Views verknüpft sind) - geosbasisdaten.ch Einträge mit einem Link zu einem Geodienst oder einer nutzbaren Datendatei - Schliesslich werden die Modelle aufgrund ihrer Komplexität ausgewählt.

NOTE :

- Der Katalog von geobasisdaten.ch enthält **346** Einträge (02.10.2023)
- **278/346** Einträge enthalten einen Link zu einem Datamodell (.ili Datei)
- Warum enthalten nicht alle Einträge einen Link zu einem Datenmodell?
- **30/278** data models contain representation information
- Von diesen **30** Vorlagen enthielten **11** Darstellungsmodelle im Excel-Format.
- Von diesen **11** Modellen enthalten **8** einen Link zu einem Geodienst oder einer nutzbaren Datendatei.
- With regard to geobasisdaten.ch, it might be appropriate to add a link to the repre- sentation models (for those that exist) in the API.
- Bei Datenmodellen, für die Dienste auf map.geo.admin.ch existieren, könnte es relevant sein, einen Link zu diesen Diensten in die API aufzunehmen.
- Ebenso wie bei der Definition dieser Dienste sollten die verwendeten Darstellungsmodelle angegeben werden, indem ein Link auf das Modell oder den API-Endpunkt eingefügt wird.

--

Ergebnisse

Mit diesen Schritten können Sie die folgenden Modelle auswählen:

Behörde	Titel	id	Minimales Modell	Webseite	Vertretung	Format	GetLegendGraphic	GetStyle	Komplexität
ARE	Wohnungsbestand und Anteil der Zweitwohnsitze	202.1	Link	Link	Link	xlsx	Link	Link	-
ASTRA	Ausrichtung von Nationalstraßen	88.1	Link	Link	Link	xlsx	Link	Link	-
ASTRA	Straßenverkehrszählung - Hauptstraßennetz	13.1;14.1	Link	Link	Link	xlsx	Link	Link	-
ASTRA	Bundesinventar historischer Verkehrswege	16.1;17.1	Link	Link	Link	xlsx	Link	Link	+
ASTRA	Sachplan Verkehr Teil Infrastruktur Straße	72.1	Link	Link	Link	xlsx	Link	Link	+
ASTRA	Netz der Hauptstraßen	90.1	Link	Link	Link	xlsx	Link	Link	-
ASTRA	Nationalstraßen	86.1	Link	Link	Link	xlsx	Link	Link	+
ARE	Reservierte Bereiche	76.1	Link	Link	Link	xlsx	Link	Link	+

Geocat

Geocat ermöglicht es auch, Informationen über MGDM abzurufen.

Man kann z.B. über die folgende Url die Datensätze filtern, die den Modellen entsprechen. However, this does not provide links to related geoservices and representation models.

Eine weitere Möglichkeit besteht darin, eine Anfrage an den CSW-Service zu senden:

```
https://www.geocat.ch/geonetwork/srv/fre/csw?service=CSW&version=2.0.2&request=GetRecords&constraintLanguage=CQL_TEXT&constraint=dc:type =
'model'&constraint_language_version=1.1.0&typeNames=csw:Record&resultType=results&ElementSetName=full&maxRecords=3&lang=de
```

aber nicht alle Datensätze entsprechen MGDM und der Filter auf dem dct:abstract LIKE 'Minimales Geodatenmodell%' scheint nicht zu funktionieren.

Durch die Verwendung der mit der Operation GetRecordById erhaltenen ID der Datensätze und des Schemas http://www.opengis.net/cat/csw/2.0.2 ist es dann möglich, die erweiterten Informationen zu erhalten, die mit dem Datensatz verbunden sind:

```
https://www.geocat.ch/geonetwork/srv/fre/csw?service=CSW&version=2.0.2&request=GetRecordById&id=f4aabaac-910e-45e9-8262-1f6f72920b99&elementSetName=full&outputSchema=http://www.opengis.net/cat/csw/2.0.2
```

--

Es ist jedoch nicht möglich, die URLs anderer verlinkter Datensätze zu erhalten, die Links zu Geodiensten enthalten.

Eine letzte Möglichkeit liegt in der Geonetwork API. Es ist möglich, den folgenden Endpunkt zu verwenden:

`https://www.geocat.ch/geonetwork/doc/api/index.html#/search/search`

und ordnen Sie es einem der folgenden 2 JSONs zu:

```
{
  "_source": {
    "includes": [
      "uuid",
      "resourceTitleObject.langfre",
      "resourceAbstractObject.langfre".
    ]
  },
    "from": 0,
   "Größe": 26,
    "query": {
      "bool": {
        "Muss": [
          {
            "query_string": {
              "query": "(any.langfre:(minimales Geodatenmodell MGDM) OR any.common:(minimales
Geodatenmodell MGDM) OR resourceTitleObject.langfre:(minimales Geodatenmodell MGDM)^2 OR
resourceTitleObject.\*:\"minimales Geodatenmodell MGDM"^6)AND (resourceType:\"service")",
              "default_operator": "AND".
            }
          }
        ]
      }
    },
    "track_total_hits": true,
    "sort":{"_id": "asc"}
  }
{
  "_source": {
    "includes": [
      "uuid",
      "resourceTitleObject.langfre",
      "resourceAbstractObject.langfre".
    ]
  },
  "from": 0,
  "size":1,
  "query": {
    "bool": {
      "Muss": [
        {
          "query_string": {
            "query": "(uuid:\c:fce3b347-cc58-4b29-bb87-a35eed4487ea\c:fce3b347-
            cc58-4b29-bb87-a35eed4487ea)", "default_operator": "AND"
          }
        }
      ]
    }
  },
  "track_total_hits": true, "sort":{"_id":
  "asc"}
  }
```

aber dadurch können die Informationen a u c h nicht verknüpft werden.

Schlussfolgerung

Wie später in der Projektdokumentation erläutert wird, ist es zur Definition der VIEWs und der Darstellungsmodelle in INTERLIS 2 notwendig, die Informationen in den Datenmodellen mit den Geodiensten oder zugehörigen Datensätzen ("flat models") und den Darstellungsmodellen (Metadaten, EXCEL- oder WMS GetStyle-Modelle) querzuverweisen, doch weder Geocat noch geobasisdaten.ch erlauben derzeit dieses Querverweisen.

8.2. Definition eines Markers in INTERLIS

Beispiel[89] für die Definition eines Markers in INTERLIS

```
<?xml version="1.0" encoding="UTF-8"?>.
<!-- File Point_Graphics_Signatures.xtf 2024-03-22 -->
<ili:transfer xmlns:ili="http://www.interlis.ch/xtf/2.4/INTERLIS"
  xmlns:geom="http://www.interlis.ch/geometry/1.0"
  xmlns="http://www.interlis.ch/xtf/2.4/StandardSymbology"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <ili:headersection>
      <ili:models>
         <ili:model>Point_Graphics</ili:model>
      </ili:models>
      <ili:sender>HEIG-VD</ili:sender>.
      <ili:comment>Point_Graphics Signatures</ili:comment>
   </ili:headersection>.
   <ili:datasection>
      <StandardSigns ili:bid="Point_Graphics">.
         <!-- Color Library -->
         <Color ili:tid="1">
            <Name>white</Name>.
            <L>100.0</L>
            <C>0.0</C>
            <H>0.0</H>
            <T>1.0</T>
         </Color>
         <Color ili:tid="2">
            <Name>schwarz</Name>
            <L>0.0</L>
            <C>0.0</C>
            <H>0.0</H>
            <T>1.0</T>
         </Color>
         <!-- Internal Symbol Font "Symbols" -->
         <Font ili:tid="10">
            <Name>Symbole</Name>
            <Internal>true</Internal>.
```

[89] https://github.com/MediaComem/FGDM4GS/blob/main/WP2-WP3/model/symbology/Point_Graphics_Signatures.xtf

```
    <Type>Symbol</Type>
  </Font>
<!-- Font/Symbol Library -->
<!-- Square -->
<FontSymbol ili:tid="101">
  <Name>Square</Name>
  <Geometrie>
<FontSymbol_Oberfläche>.
      <Geometrie>
          <geom:Fläche>
           <geom:exterior>
             <geom:polyline>
               <geom:coord>
                 <geom:c1>-0.5</geom:c1><geom:c2>-0.5</geom:c2>.
               </geom:coord>
               <geom:coord>
                 <geom:c1>0.5</geom:c1><geom:c2>-0.5</geom:c2>.
               </geom:coord>
               <geom:coord>
                 <geom:c1>0.5</geom:c1><geom:c2>0.5</geom:c2>
               </geom:coord>
               <geom:coord>
                 <geom:c1>-0.5</geom:c1><geom:c2>0.5</geom:c2>.
               </geom:coord>
               <geom:coord>
                 <geom:c1>-0.5</geom:c1><geom:c2>-0.5</geom:c2>.
               </geom:coord>
            </geom:polyline>.
           </geom:exterior>
         </geom:surface>
      </Geometry>.
   </FontSymbol_Surface>.
  </Geometry>.
  <Geometrie>
    <FontSymbol_Polyline>.
      <Color ili:ref="2"></Color>.
      <Geometrie>
        <geom:polyline>
          <geom:coord>
            <geom:c1>-0.5</geom:c1><geom:c2>-0.5</geom:c2>.
          </geom:coord>
          <geom:coord>
            <geom:c1>0.5</geom:c1><geom:c2>-0.5</geom:c2>.
          </geom:coord>
          <geom:coord>
```

```
                    <geom:c1>0.5</geom:c1><geom:c2>0.5</geom:c2>
                  </geom:coord>
                  <geom:coord>
                    <geom:c1>-0.5</geom:c1><geom:c2>0.5</geom:c2>.
                  </geom:coord>
                  <geom:coord>
                    <geom:c1>-0.5</geom:c1><geom:c2>-0.5</geom:c2>.
                  </geom:coord>
                </geom:polyline>.
              </Geometry>.
            </FontSymbol_Polyline>.
          </Geometry>.
          <Font ili:ref="10"></Font>.
        </FontSymbol>
        <!-- Kreis -->
        <FontSymbol ili:tid="102">
          <Name>Circle</Name>.
          <!-- Hier wird kein FontSymbol_Surface definiert, da dies für einen Kreis nicht
          notwendig ist und auch bedeuten würde, dass eine große Anzahl von Koordinaten
          definiert werden müsste. -->
          <Geometrie>
            <FontSymbol_Polyline>.
              <Color ili:ref="2"></Color>.
              <Geometrie>
                <geom:polyline>
                  <geom:coord>
                    <geom:c1>0.0</geom:c1><geom:c2>0.5</geom:c2>.
                  </geom:coord>
                  <geom:arc>
                    <geom:c1>0.0</geom:c1><geom:c2>0.0</geom:c2>.
                    <geom:a1>0.0</geom:a1><geom:a2>6.283185307179586</geom:a2><geom:r>0.5</geom:r
>
                  </geom:arc>
                </geom:polyline>.
              </Geometry>.
            </FontSymbol_Polyline>.
          </Geometry>.
          <Font ili:ref="10"></Font>.
          </FontSymbol>
        <!-- Dreieck -->
        <FontSymbol ili:tid="103">
            <Name>Dreieck</Name>
            <Geometrie>
              <FontSymbol_Oberfläche>.
                <Geometrie>
                  <geom:Fläche>
```

```
            <geom:exterior>
              <geom:polyline>
                <geom:coord>
                  <geom:c1>-0.5</geom:c1><geom:c2>-0.5</geom:c2>.
                </geom:coord>
                <geom:coord>
                  <geom:c1>0.0</geom:c1><geom:c2>0.5</geom:c2>.
                </geom:coord>
                <geom:coord>
                  <geom:c1>0.5</geom:c1><geom:c2>-0.5</geom:c2>.
                </geom:coord>
                <geom:coord>
                  <geom:c1>-0.5</geom:c1><geom:c2>-0.5</geom:c2>.
                </geom:coord>
             </geom:polyline>.
            </geom:exterior>
          </geom:surface>
        </Geometry>.
      </FontSymbol_Surface>.
    </Geometry>.
    <Geometrie>
      <FontSymbol_Polyline>.
        <Color ili:ref="2"></Color>.
        <Geometrie>
          <geom:polyline>
            <geom:coord>
              <geom:c1>-0.5</geom:c1><geom:c2>0.0</geom:c2>.
            </geom:coord>
            <geom:arc>
              <geom:c1>0.5</geom:c1><geom:c2>0.0</geom:c2>.
              <geom:a1>0.0</geom:a1><geom:a2>0.5</geom:a2><geom:r>0.5</geom:r>
            </geom:arc>
            <geom:arc>
              <geom:c1>-0.5</geom:c1><geom:c2>0.0</geom:c2>.
              <geom:a1>0.0</geom:a1><geom:a2>-0.5</geom:a2><geom:r>0.5</geom:r>
            </geom:arc>
          </geom:polyline>.
        </Geometry>.
      </FontSymbol_Polyline>.
    </Geometry>.
    <Font ili:ref="10"></Font>.
  </FontSymbol>
  <!-- Star -->
  <!-- Beachten Sie, dass dieses Symbol komplexer gestaltet werden konnte -->.
  <FontSymbol ili:tid="104">
```

```
 <Name>Star</Name>
 <Geometrie>
<FontSymbol_Oberfläche>.
     <Geometrie>
       <geom:Oberfläche>
         <geom:exterior>
           <geom:polyline>
             <geom:coord>
               <geom:c1>-4.0</geom:c1><geom:c2>0.0</geom:c2>.
             </geom:coord>
             <geom:coord>
               <geom:c1>-1.0</geom:c1><geom:c2>-1.0</geom:c2>.
             </geom:coord>
             <geom:coord>
               <geom:c1>0.0</geom:c1><geom:c2>-4.0</geom:c2>.
             </geom:coord>
             <geom:coord>
               <geom:c1>1.0</geom:c1><geom:c2>-1.0</geom:c2>.
             </geom:coord>
             <geom:coord>
               <geom:c1>4.0</geom:c1><geom:c2>0.0</geom:c2>.
             </geom:coord>
             <geom:coord>
               <geom:c1>1.0</geom:c1><geom:c2>1.0</geom:c2>
             </geom:coord>
             <geom:coord>
               <geom:c1>0.0</geom:c1><geom:c2>4.0</geom:c2>.
             </geom:coord>
             <geom:coord>
               <geom:c1>-1.0</geom:c1><geom:c2>1.0</geom:c2>.
             </geom:coord>
             <geom:coord>
               <geom:c1>-4.0</geom:c1><geom:c2>0.0</geom:c2>.
             </geom:coord>
           </geom:polyline>.
         </geom:exterior>
       </geom:surface>
     </Geometry>.
  </FontSymbol_Surface>.
 </Geometry>.
 <Geometrie>
  <FontSymbol_Polyline>.
     <Color ili:ref="2"></Color>.
     <Geometrie>
       <geom:polyline>
```

```
            <geom:coord>
              <geom:c1>-4.0</geom:c1><geom:c2>0.0</geom:c2>.
            </geom:coord>
            <geom:coord>
              <geom:c1>4.0</geom:c1><geom:c2>0.0</geom:c2>.
            </geom:coord>
          </geom:polyline>.
        </Geometry>.
      </FontSymbol_Polyline>.
      <FontSymbol_Polyline>.
        <Color ili:ref="2"></Color>.
        <Geometrie>
          <geom:polyline>
            <geom:coord>
              <geom:c1>0.0</geom:c1><geom:c2>-4.0</geom:c2>.
            </geom:coord>
            <geom:coord>
              <geom:c1>0.0</geom:c1><geom:c2>4.0</geom:c2>.
            </geom:coord>
          </geom:polyline>.
        </Geometry>.
      </FontSymbol_Polyline>.
    </Geometry>.
    <Font ili:ref="10"></Font>.
  </FontSymbol>
  <!-- Cross -->
  <FontSymbol ili:tid="105">
    <Name>Cross</Name>.
    <Geometrie>
      <FontSymbol_Polyline>.
        <Color ili:ref="2"></Color>.
        <Geometrie>
          <geom:polyline>
            <geom:coord>
              <geom:c1>-0.5</geom:c1><geom:c2>0.0</geom:c2>.
            </geom:coord>
            <geom:coord>
              <geom:c1>0.5</geom:c1><geom:c2>0.0</geom:c2>.
            </geom:coord>
          </geom:polyline>.
        </Geometry>.
      </FontSymbol_Polyline>.
      <FontSymbol_Polyline>.
        <Color ili:ref="2"></Color>.
        <Geometrie>
```

```
                <geom:polyline>
                  <geom:coord>
                    <geom:c1>0.0</geom:c1><geom:c2>-0.5</geom:c2>.
                  </geom:coord>
                  <geom:coord>
                    <geom:c1>0.0</geom:c1><geom:c2>0.5</geom:c2>.
                  </geom:coord>
                </geom:polyline>.
              </Geometry>.
            </FontSymbol_Polyline>.
          </Geometry>.
          <Font ili:ref="10"></Font>.
        </FontSymbol>
        <!-- X -->
        <!-- Es ist nicht notwendig, dieses Symbol zu definieren, da es durch Drehen eines
        Kreuzes erreicht werden kann -->.
      <!-- Symbol Signs -->
      <SymbolSign ili:tid="2001">
        <ili:Name>Symbol</ili:Name>
        <Scale>1.0</Scale>
        <Rotation>45</Rotation>
        <Color ili:ref="2"></Color>.
        <Symbol ili:ref="105"></Symbol>.
      </SymbolSign>
    </StandardSigns>
  </ili:datasection>
</ili:transfer>
```