

FGDM4GS

« Flat Geodata Models for Geoservices »

Approche basée modélisation conceptuelle pour la mise en œuvre de géoservices pour les MGDM (attributs et symbologie)



Auteurs : Maxime Collombin, Olivier Ertz, Jens Ingensand

Date : Oct. 2024

Table des matières

1.	Contexte et problématique.....	3
2.	Objectifs du projet.....	7
3.	Sélection de géodonnées de base.....	8
3.1.	Méthodologie.....	8
3.2.	Inventaire des modèles sélectionnés.....	9
3.3.	Recommandations.....	11
4.	Définition de vues dans le modèle conceptuel.....	12
4.1.	Méthodologie.....	12
4.2.	Types de VIEW.....	13
4.3.	Exemples sur la base des modèles sélectionnés.....	16
4.4.	Recommandations.....	24
5.	Modèle de représentation.....	25
5.1.	Méthodologie.....	26
5.2.	Comparaison INTERLIS et OGC.....	27
5.3.	Classe d'exigences « Core ».....	28
5.4.	Classe d'exigences « Basic Vector Features Styling ».....	33
5.5.	Classe d'exigences « Labeling».....	41
5.6.	Classe d'exigences « Advanced strokes».....	43
5.7.	Classe d'exigences « Advanced fills».....	47
5.8.	Classe d'exigence pour les expressions additionnelles.....	49
5.9.	Exemples sur la base des modèles sélectionnés.....	51
5.10.	Recommandations.....	57
6.	Automatisation de la création de géoservices.....	59
7.	Conclusion.....	59
8.	Annexes.....	61
8.1.	Procédure de sélection des modèles données pour cette étude.....	61
8.2.	Définition d'un marker en INTERLIS.....	64

1. Contexte et problématique

Au sens de l'art. 3¹ de la Loi fédérale sur la géoinformation (LGéo), on entend par « géodonnées de base », les géodonnées qui se fondent sur un acte législatif fédéral, cantonal ou communal. De plus, selon l'art. 9² de l'Ordonnance sur la géoinformation (OGéo), le service spécialisé compétent de la Confédération prescrit un modèle de géodonnées minimal (MGDM). Il y fixe la structure et le degré de spécification du contenu. Et selon le document du Plan de mise en œuvre³, l'OGéo « *fixe un délai de cinq ans⁴ pour la mise à disposition des géodonnées de base sous la forme prescrite par les MGDM, à compter de la date d'existence de ces derniers* » et souligne qu'il ressort « *de divers retours émanant des cantons que le respect du délai exigé n'est pas réaliste pour tous les cantons et tous les thèmes* », ce pourquoi un « *Plan de mise en œuvre* »⁵ a été élaboré. Cette mise en œuvre coordonnée vise à établir des priorités parmi les géodonnées de base mises à disposition sur l'infrastructure d'agrégation *geobasisdaten.ch* tout en respectant au mieux les délais impartis et couvrant l'intégralité du territoire suisse.

La section « *Programmes de mise en œuvre en cours* » du plan de mise en œuvre de la Conférence des services cantonaux de la Géoinformation et du Cadastre (CGC) situe l'état d'avancement des programmes afférents. Il résulte de l'expérience acquise dans le cadre de la mise en œuvre du thème des cartes des dangers que des schémas d'application, respectivement des modèles logiques doivent être définis préalablement pour la mise en œuvre conforme au modèle des géodonnées de base dans le cas de la méthode d'accès direct. C'est pourquoi l'organe de coordination de la géoinformation au niveau fédéral (GCS) et la CGC ont défini des instructions pour l'échange de géodonnées conformes à un modèle (MDX)⁶ dans le cadre d'un projet conjoint en 2016. Ces instructions guident les services compétents lors de la réalisation de services de téléchargement conformes aux modèles prescrits, comme la loi sur la géoinformation l'exige.

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_FR.pdf

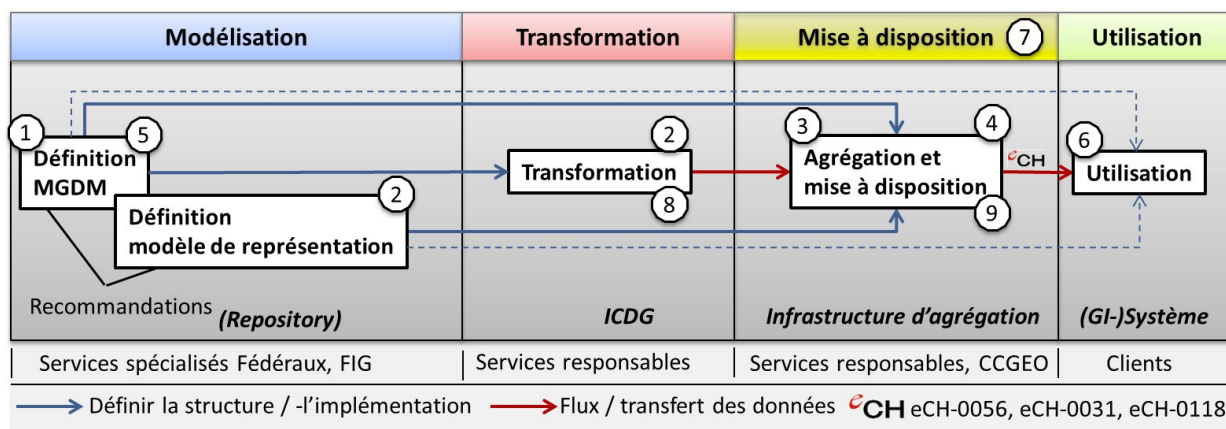


Figure 1: aperçu du processus de mise en oeuvre des géodonnées de base extrait du Plan de mise en oeuvre des géodonnées de base relevant du droit fédéral dont la compétence est attribuée aux cantons

En parallèle au MDX, une « *Recommandation pour le développement de modèles de représentation* »⁷ a également été définie afin d'homogénéiser les représentations cartographiques dans les services de visualisation.

Une fois leur mise en œuvre terminée, les géodonnées de base conformes aux modèles (MGDM) sont publiées sur l'infrastructure d'agrégation des cantons (IA) *geodienste.ch* pour celles relevant de la compétence cantonale (voir Figure 1) et sur l'infrastructure fédérale de géodonnées (IFDG) via les services de visualisation de swisstopo pour celles relevant de la compétence fédérale.

Les informations relatives à la configuration des géoservices de l'IA sont documentées dans des rapports pouvant être consultés dans la section « *Bases de mise en œuvre des cantons pilotes* » sur la page de la CGC : « *Processus de mise en œuvre* »⁸ ou via les ressources liées à l'enregistrement *geocat*⁹ correspondant au flux *geodienste.ch* (Atom Feed / OpenSearch Service) comme il est le cas dans l'exemple ci-dessous (voir Table 1) où les attributs du WMS sont détaillés pour le MGDM Zones Réservées (ID 76)¹⁰.

⁷ <https://drive.switch.ch/index.php/apps/files/?dir=/mediamaps/01-Projets/02-En-cours/01-Flache-Geodatenmod-elle/Références&fileid=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

Table 1: extrait du rapport de mise en oeuvre pour le MGDM Zones Réservées (ID 76) indiquant les attributs du modèle conceptuel devant être obtenus via l'opération GetFeatureInfo du wms

Si les informations relatives à la configuration des géoservices peuvent être consultées pour les géodonnées de base mises en œuvre, aucune documentation n'est en revanche disponible pour les géoservices de l'IFDG.

Malgré l'existence des instructions pour l'échange de géodonnées conformes à un modèle (MDX), des différences sont tout de même constatées entre les données et géoservices publiées par les cantons et ceux de l'IA. C'est le cas notamment dans l'exemple ci-dessous (voir Figure 2: différences entre les géoservices *vsgis.ch* et *geodienste.ch*) effectuant la comparaison entre deux différents géoservices, à savoir celui de *geodienste.ch*¹¹ (à droite) et celui de *vsgis.ch*¹² (à gauche) portant tous deux sur le MGDM Zones Réservées (ID 76) pour lesquels à la fois les attributs exposés et la représentation ne correspondent pas. A des fins de démonstration, les 2 géoservices ont été importés sur *map.geo.admin.ch* et peuvent être consultés à l'adresse suivante : <https://s.geo.admin.ch/bwgnkx2ut9k>.

¹¹ https://geodienste.ch/db/planungszonen_v1_1_0/fra?SERVICE=WMS&REQUEST=GetCapabilities

¹² https://map.vsgis.ch/valdebagnes/wsgi/mapserv_proxy?ogcserver=source+for+None&SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities

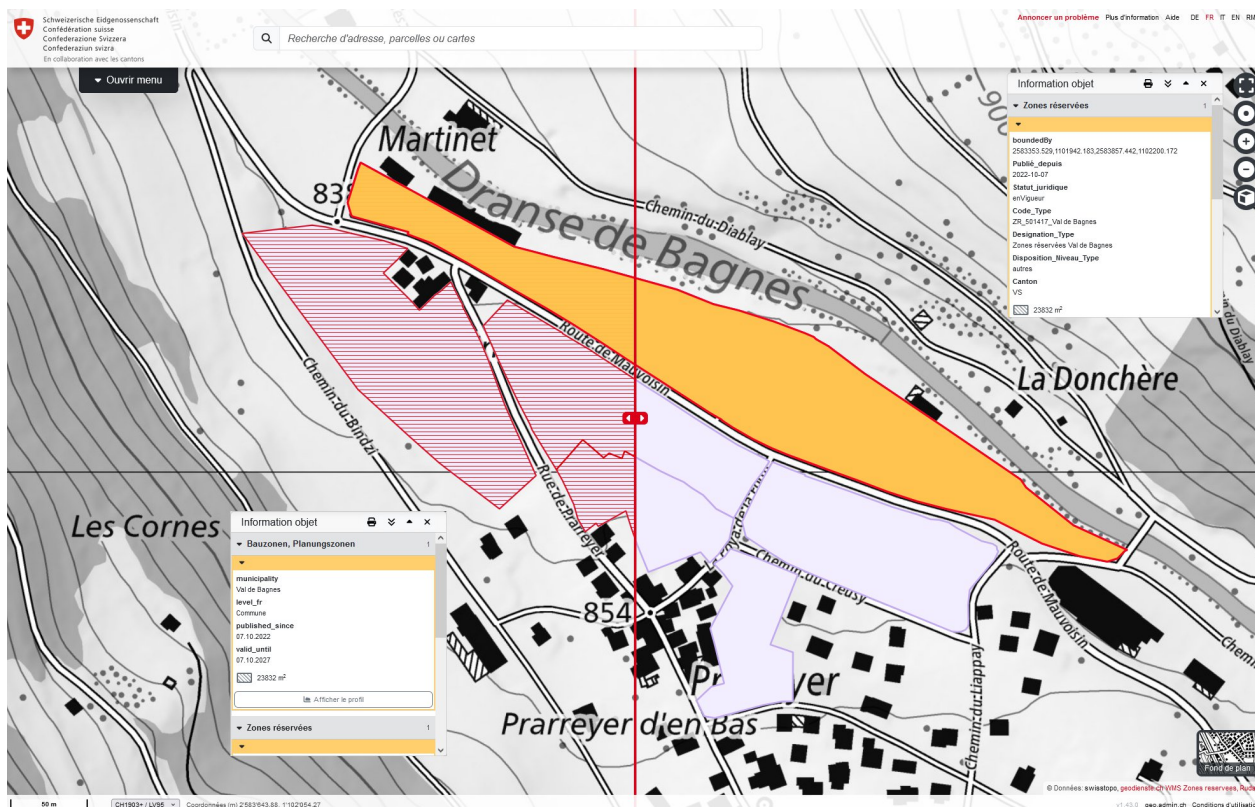


Figure 2: différences entre les géoservices vsgis.ch et geodienst.ch

Nb. si vsgis.ch n'est pas l'instance officielle de publication des géodonnées du canton du Valais, il présente l'avantage de respecter les standards OGC et donc une intégration sur d'autres instances telles que map.geo.admin.ch. La cohérence avec les données du canton du Valais¹³ a toutefois été vérifiées au préalable (voir Figure 3).

Fields:	Drawing Info:
- 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)	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:

Figure 3: attributs et symbologie de l'API REST ArcGISOnline du canton du Valais

¹³ <https://sit.vs.ch/arcgis/rest/services/PAZ/MapServer/0>

Les problèmes qui résultent de ces constats sont les suivants.

- d'abord pour les utilisateurs professionnels des infrastructures géoservices : il existe une hétérogénéité entre les services web des différents cantons et de la confédération - à la fois au niveau de la publication des attributs, mais également au niveau de la représentation des géodonnées. La mise en commun de plusieurs services dans un seul projet peut engendrer des incohérences
- ensuite pour les gestionnaires d'infrastructures de géoservices : bien qu'il existe pour certains MGDM une documentation qui décrit à la fois les attributs à inclure dans un service web et également la représentation des données, cette documentation n'est souvent pas bien respectée ou difficile à exploiter.

Nous pouvons ainsi formuler l'hypothèse qu'une méthode plus normative favorisant même des automatisations pourrait être bénéfique à la fois pour la CGC (agrégation de données sur geodienste) et également pour chaque canton ou entité qui publie des données relatives à un MGDM, et in-fine les utilisateurs professionnels des infrastructures géoservices.

Ce projet se propose d'explorer des pistes pour exploiter au plus près les standards en vigueur dans la perspective de publication des MGDM sous la forme de géoservices interopérables. Ceci à la fois pour définir une structure dénormalisée / à plat d'un modèle minimal de géodonnées qu'une description symbolique.

2. Objectifs du projet

Ce projet **FGDM4GS**, pour « Flat Geodata Models for Geoservices », vise à explorer les possibilités offertes par la modélisation conceptuelle pour la mise en œuvre de géoservices se basant sur des géodonnées de bases et les prescriptions des MGDM associés. Afin de favoriser la mise en œuvre interopérable de géoservices, il s'agit d'évaluer :

- les possibilités de définition de VIEW INTERLIS pour la publication de données attributaires (avec mise en exemple sur la base d'une sélection de modèles)
- les capacités graphiques SYMBOLOGY INTERLIS pour la publication de représentations cartographiques (avec mise en exemple sur la base d'une sélection de modèles)

Au terme de chaque analyse, des perspectives et recommandations sont établies sur la base des résultats afin de pouvoir servir de base de réflexion pour une possible modification des bases normatives et outils liés. En conséquence, les destinataires des résultats de ce projet et de ces recommandations sont la CGC/KGK et COSIG/KOGIS, ainsi que les groupes eCH Geoinformation et OSIG/SOGI GGMM.

3. Sélection de géodonnées de base

3.1. Méthodologie

Dans cette phase préliminaire il s'agit de sélectionner des géodonnées de base pour tenter d'exemplifier les concepts évalués dans ce travail, d'un point de vue dénormalisation mais aussi symbolique. Pour cela nous choisirons des modèles pour lesquels un lien peut être établi avec un géoservice existant, afin d'en évaluer la cohérence avec les modèles de données et de représentation mis en œuvre. Les informations obtenues via le géoservice permettent également de disposer de toutes les informations pour la définition de la VIEW et du modèle de représentation.

Cette sélection se base sur une analyse à partir des entrées de *geobasisdaten.ch* afin de déterminer dans quelle mesure les géodonnées de base :

- ... possèdent un modèle de données conceptuel
- ... contiennent l'élément syntaxique « VIEW » du langage INTERLIS permettant de décrire des vues comme détaillé dans le chapitre 4
- ... respectent la définition de modèles de représentation tel que préconisé par les recommandations sur l'élaboration de modèles de représentation pour les MDGM¹⁴
- ... associent les MDGM à des géoservices
 - afin de valider la cohérence de leur mise en œuvre vis à vis des modèles conceptuels et des informations de représentation
 - ou dans le cas contraire obtenir des informations complémentaires pour la définition de VIEW et modèles de représentation

Pour ce faire, une utilisation croisée de l'API *geobasisdaten.ch*¹⁵ et du service CSW¹⁶ de *geocat.ch*¹⁷ est réalisée. Le schéma suivant illustre le concept de sélection de MDGM.

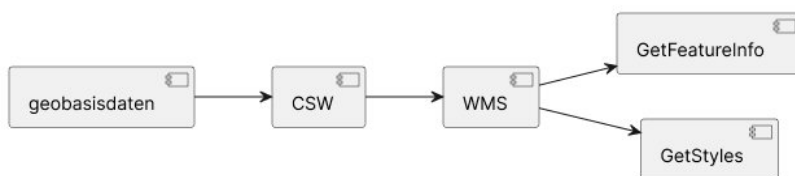


Figure 4: concept de mise en relation des MDGM avec les géoservices

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=CSW&version=2.0.2&request=GetCapabilities>

¹⁷ <https://www.geocat.ch/geonetwork/srv/fre/catalog.search#/home>

Les informations de *geobasisdaten.ch* sont mises en relation avec les géoservices associés en sollicitant le service CSW de *geocat.ch* afin d'obtenir les attributs à intégrer dans les VIEWS afin de pouvoir configurer convenablement :

- La requête GetFeatureInfo sur les services WMS
- La représentation cartographique ainsi que l'opération GetStyles

L'analyse a été effectuée au travers de différents scripts accessibles au niveau de ce dépôt GitHub¹⁸ (voir procédure de sélection en annexe).

3.2. Inventaire des modèles sélectionnés

L'API *geobasisdaten.ch* contenait 347 entrées au 14.02.2023 dont le 80% soit 283/347 entrées comportaient un lien vers un modèle de données (.ili) :

- Aucun des modèles existants ne comporte de « VIEW ». Ce qui confirme que ce n'est pas dans les usages.
- Un nombre inférieur à 5% soit environ 10/347 entrées de l'API contient un lien vers des informations de représentation au format tabulaire préconisé. 33%, soit 115/347 entrées ont pu être reliées à des géoservices.

Finalement, une sélection de MGDm est effectuée en tenant compte des critères précédemment évoqués et de la complexité interne des modèles.

Il en résulte les données dans le tableau ci-dessous sur lesquels nous allons nous baser. Il met en relation les MGDm avec les modèles de données ainsi que les géoservices leur étant associés. Les modèles contiennent tous les attributs nécessaires à l'opération GetFeatureInfo mais ces derniers sont répartis dans différentes classes. Il est à noter que l'opération GetStyles n'y est pas considérée car la réponse obtenue non satisfaisante.

Notons que les enregistrements de l'API *geobasisdaten.ch* comportent un lien vers *geocat* mais uniquement vers les ressources de type : « Modèle de données » comme dans l'exemple ci-dessous (voir figure 5) pour la géo-donnée de base « Zones Réservées ».

On peut également constater que la plateforme *geocat.ch* ne lie pas les ressources de type : « Modèle de données » aux ressources de type « Service ».

Il en est de même pour le géoservice *geodienst.ch* qui ne comporte aucune information vers le modèle de donnée, *geobasisdaten.ch* ou encore les métadonnées *geocat.ch*.

¹⁸ <https://github.com/MediaComem/FGDM4GS/tree/main/WP1/T1-1>

Nom	Modèle	WMS GetFeatureInfo	Rapport
Axes des routes nationales (86.1 ¹⁹)	Lien ²⁰	Lien ²¹	-
Inventaire fédéral des voies de communication historiques (16.1 ²² , 17.1 ²³)	Lien ²⁴	Lien ²⁵	-
Zones réservées (76.1 ²⁶)	Lien ²⁷	Lien ²⁸	Lien ²⁹
Plan sectoriel des transports Partie route (72.1) ³⁰	Lien ³¹	Lien ³²	-

Table 2: inventaire des modèles sélectionnés

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=WMS&VERSION=1.3.0&REQUEST=GetFeatureInfo&LAYERS=ch.astra.nationalstrassenachsen&QUERY_LAYERS=ch.astra.nationalstrassenachsen&CRS=EPSG:2056&BBOX=2531423.89,1155079.22,2532223.89,1155679.22&FEATURE_COUNT=1&HEIGHT=2&WIDTH=2&FORMAT=image/png&INFO_FORMAT=text/plain&I=1&J=1&lang=fr

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=WMS&VERSION=1.3.0&REQUEST=GetFeatureInfo&LAYERS=ch.astra.ivs-nat&QUERY_LAYERS=ch.astra.ivs-nat&CRS=EPSG:2056&BBOX=2531423.89,1155079.22,2532223.89,1155679.22&FEATURE_COUNT=1&HEIGHT=2&WIDTH=2&FORMAT=image/png&INFO_FORMAT=text/xml&I=1&J=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=WMS&VERSION=1.3.0&REQUEST=GetFeatureInfo&CRS=EPSG:3857&BBOX=742465.67788010137155652,5902893.5199219873175025,751489.7654410689137876,5913919.95374835748225451&WIDTH=667&HEIGHT=815&LAYERS=planungszone&FORMAT=image/png&QUERY_LAYERS=planungszone&INFO_FORMAT=application/vnd.ogc.gml&I=500&J=267&FEATURE_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=WMS&VERSION=1.3.0&REQUEST=GetFeatureInfo&LAYERS=ch.astra.sachplan-infrastrukturstrasse_kraft&QUERY_LAYERS=ch.astra.sachplan-infrastrukturstrasse_kraft&CRS=EPSG:2056&BBOX=2531423.89,1155079.22,2532223.89,1155679.22&FEATURE_COUNT=1&HEIGHT=2&WIDTH=2&FORMAT=image/png&INFO_FORMAT=text/plain&I=1&J=1



Figure 5: modèle de géodonnées *Planungszonen_V1_1*³³

3.3. Recommandations

3.3.1. Liens vers tous les types de ressources geocat.ch

Intégrer les liens vers tous les types de ressources *geocat.ch* (« Modèle de données », « Service », « Données ») au niveau de *geobasisdaten.ch* lorsque disponibles.

3.3.2. Intégration de liens ressources sur geocat.ch

Intégrer des liens entre les différents types de ressources *geocat.ch* (« Modèle de données », « Service », « Données ») lorsque disponibles.

3.3.3. Références aux données, géoservices et métadonnées dans les modèles

Les modèles *.ili ne comportent aucune référence ni aux données ni aux géoservices associés ni encore aux métadonnées. Les références pourraient cependant être ajoutées au travers d'un méta-attribut ou d'une classe ou structure spécifique directement dans le modèle tel qu'il est le cas les modèles cadres³⁴ du cadastre RDPPF.

3.3.4. Intégration des métadonnées dans les services

Les géoservices et les données ne comportent ni de référence aux modèles, ni aux métadonnées. Ces informations pourraient être ajoutées aux exigences de eCH-0056.

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

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

4. Définition de vues dans le modèle conceptuel

L'art. 10³⁵ de l'OGéo mentionne que le langage de description des MGDМ doit correspondre à une norme reconnue. L'art. 5³⁶ de l'OGéo swisstopo, lui, précise que le langage général de description de modèles de géodonnées doit notamment respecter la norme eCH-0031 INTERLIS 2 – Manuel de référence³⁷. Pour satisfaire le premier objectif de ce projet, nous évaluons donc les possibilités de création de vues avec INTERLIS.

Une « VIEW » est une règle intervenant dans la grammaire du langage INTERLIS permettant de sélectionner des attributs à partir de classes ou associations de modèles de données. Le concept de « VIEW » est défini dans le manuel de référence d'INTERLIS (refhb24) à la section « 2.15. Sichten ». Nous faisons ici référence à la version allemande du manuel de référence d'INTERLIS (refhb24) car il en existe une version dans le format asciidoc publiée sur un dépôt GitHub³⁸ associé au compte de Geostandards.ch. Cela permet ainsi d'en référencer les sections.

```
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.
```

Bloc de code 1: extrait de grammaire BNF d'INTERLIS définissant une VIEW

Afin de ne pas altérer les modèles de base, les exemples de VIEW des sections suivantes sont définies dans des modèles dérivés, c'est à dire qui dépendent des modèles existants.

4.1. Méthodologie

Les sections suivantes explorent dans quelle mesure des « VIEW » peuvent être utiles pour la configuration de jeux de géodonnées ou géoservices conformes aux modèles. Pour ce faire, nous nous basons sur la documentation de la section 3.15 « Vues » du manuel de référence ainsi que sur les exemples de tests d'ili2c³⁹ de manière à bien comprendre le concept de création de VIEW et le détailler ici. Dans un second temps, les VIEW sont appliquées aux MGDМ sélectionnés et des recommandations faisant suite à nos observations sont formulées.

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. Types de VIEW

Les 5 possibilités pour définir des VIEWS en INTERLIS sont décrites ci-dessous :

- PROJECTION OF
- JOIN OF & WHERE
- UNION OF
- AGGREGATION OF
- INSPECTION OF

4.2.1. VIEW PROJECTION OF

Selon le manuel de référence, la vue projection (mot-clé PROJECTION OF) *constitue la vue la plus simple. Elle permet de visualiser l'objet de base dans le diagramme de classes UML (classe, structure, association ou vue) sous une forme modifiée (affichage d'une partie seulement des attributs et selon un ordre modifié, par exemple).* Cela peut être utile pour simplifier l'affichage des données ou pour les adapter à un contexte particulier. Par exemple, pour mettre à disposition des vues avec des intitulés traduits des attributs (ex. une vue avec les noms des attributs en allemand et une vue avec des attributs en français), il est possible de créer une vue par langue utilisant le concept « PROJECTION OF ».

```
INTERLIS 2.3;
CONTRACTED MODEL Test AT "http://www.interlis.ch/ili2c/tests/" VERSION "1"=
  FUNCTION makeConstant(text : TEXT):TEXT;
  TOPIC Base =
    STRUCTURE A =
      Attr1: TEXT*20;
    END A;
    CLASS B =
      Attr1: TEXT*20;
      Attr2: BAG OF A;
    UNIQUE
      Attr1;
    END B;
  END Base;
  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.
```

Bloc de code 2: exemple⁴⁰ de définition d'une VIEW PROJECTION OF

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

4.2.2. VIEW JOIN OF & WHERE

Selon le manuel de référence (refhb24), la vue jointure (mot-clé JOIN OF) permet de former le produit cartésien (ou produit vectoriel) des classes de base (classe ou vue), c.-à-d. qu'il existe alors autant d'objets de la classe de jonction que de combinaisons d'objets des différentes classes de base. Le fait de lui associer une clause WHERE permet de définir les champs de jonction (l'ensemble des objets-vues défini par l'intermédiaire de la loi de formation peut être soumis à des restrictions supplémentaires par l'introduction de conditions). Cela peut être utile, par exemple, pour associer des statistiques cantonales en matière de votation compte tenu de différents critères.

```
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;
    CLASS 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.
```

Bloc de code 3: exemple⁴¹ de définition d'une VIEW JOIN OF & WHERE

4.2.3. VIEW UNION OF

Selon le manuel de référence (refhb24), la vue union (mot-clé UNION OF) permet la fusion de différentes classes de base en une classe unique. Les attributs des différentes classes de base sont généralement affectés à un attribut de la vue union. Le type d'attribut de la classe de base doit être compatible avec celui de la vue union (même type ou extension de celui-ci). Cela permet, par exemple, de fusionner 2 jeux de données possédant une même structure.

ProjectionDef.ili

⁴¹ https://github.com/claeis/ili2c/blob/c506ae466333d726b885ca7fae4ce6825e94176d/test/data/ili23/view/view_AcceptBasicJoin-Def.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.

```

Bloc de code 4:exemple⁴² de définition d'une VIEW UNION OF

4.2.4. VIEW AGGREGATION OF

Selon le manuel de référence (refhb24), la vue agrégation (mot-clé AGGREGATION OF) *permet de regrouper, en une même instance, toutes les instances d'un ensemble de base ou celles présentant une identité avec la combinaison d'attributs requise*. Cela permet par exemple de calculer des statistiques sur un ensemble de données.

```

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 Base;
  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.

```

Bloc de code 5: exemple⁴³ de définition d'une VIEW AGGREGATION OF

4.2.5. VIEW INSPECTION OF

Selon le manuel de référence (refhb24), la vue inspection (mot-clé INSPECTION OF) permet d'obtenir l'ensemble de tous les éléments structurés (définis via BAG OF, LIST OF ou dans le respect d'une ligne, d'une surface simple ou d'une partition d'un territoire) appartenant à un attribut de sous-structure (direct ou indirect) d'une classe d'objets. Cela permet d'explorer une liste d'éléments pour les afficher de manière structurée.

```

INTERLIS 2.3;
MODEL Test AT "http://www.interlis.ch/ili2c/tests/" VERSION "1" =
    TOPIC Base =
        STRUCTURE 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 Base;
END Test.

```

Bloc de code 6: exemple⁴⁴ de définition d'une VIEW INSPECTION OF

4.3. Exemples sur la base des modèles sélectionnés

Les exemples ci-dessous de VIEWS pour les modèles de données sélectionnés ont été définis en INTERLIS. Ils illustrent les possibilités offertes par INTERLIS pour la définition de VIEWS et concernent les modèles de données suivants : 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_AcceptBaselineOfUnrenamedInspectionDef.ili

4.3.1. Exemple 1 : Axes des routes nationales

Le 1^{er} exemple est défini sur la base du MGDМ Axes des routes nationales (86.1⁴⁵).

La définition de la VIEW mobilise 3 différentes classes au travers de la combinaison de la vue JOIN OF et du recours à la clause WHERE. Afin de pouvoir définir le modèle graphique à la section 5, il est nécessaire de faire intervenir l'attribut supplémentaire « Type » correspondant à l'attribut « AxisType » de la classe « Axis ».

```
INTERLIS 2.3;
MODEL Axis_LV95_V1_1_d (en)
  AT "mailto:maxime.collombin@heig-vd.ch"
  VERSION "2023-12-06" =
  IMPORTS 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.
```

Bloc de code 7: modèle dérivé⁴⁶ du modèle Axis_LV95_V1_1 avec la définition de la 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

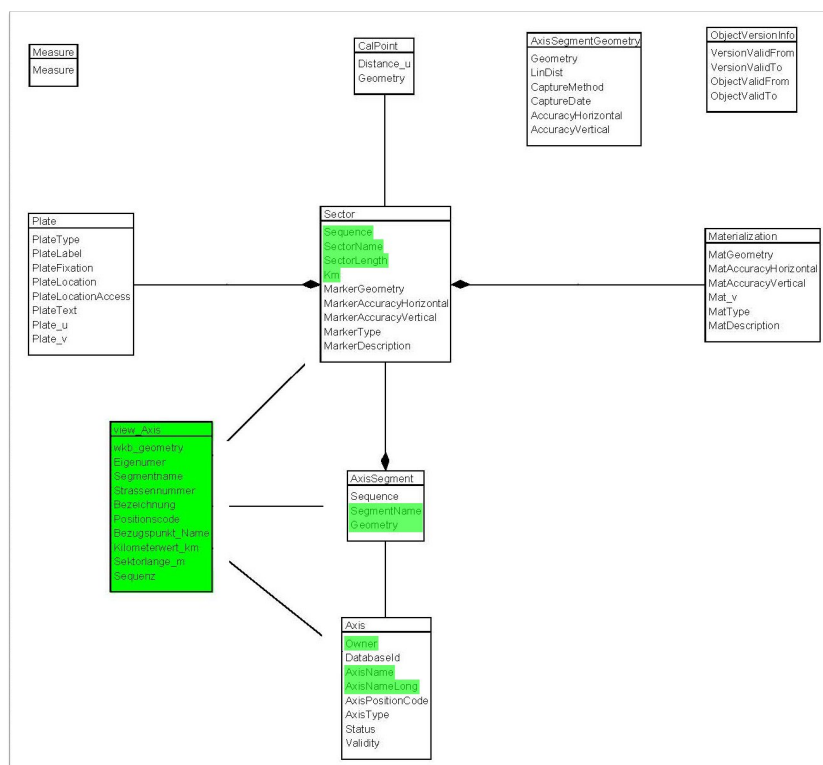


Figure 6: Diagramme de classe UML avec les classes et attributs mobilisés dans la VIEW

4.3.2. Exemple 2 : Inventaire des voies de communication historiques

Le 2^{ème} exemple est défini sur la base du MGDM Inventaire fédéral des voies de communication historiques (16.1⁴⁷, 17.1⁴⁸). Le modèle StandardSymbology⁴⁹ qui est exploré à la section 5 étant défini en INTERLIS 2.4, il est néanmoins nécessaire d'adapter les modèles de données pour qu'ils puissent être compatibles.

La définition de la VIEW mobilise 7 différentes classes au travers de la combinaison de la vue JOIN OF et du recours à la clause WHERE avec plusieurs conditions.

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" =
  IMPORTS IVS_V3;
  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.

```

Bloc de code 8: modèle dérivé⁵⁰ du modèle IVS_V2_1 avec la définition de la VIEW

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

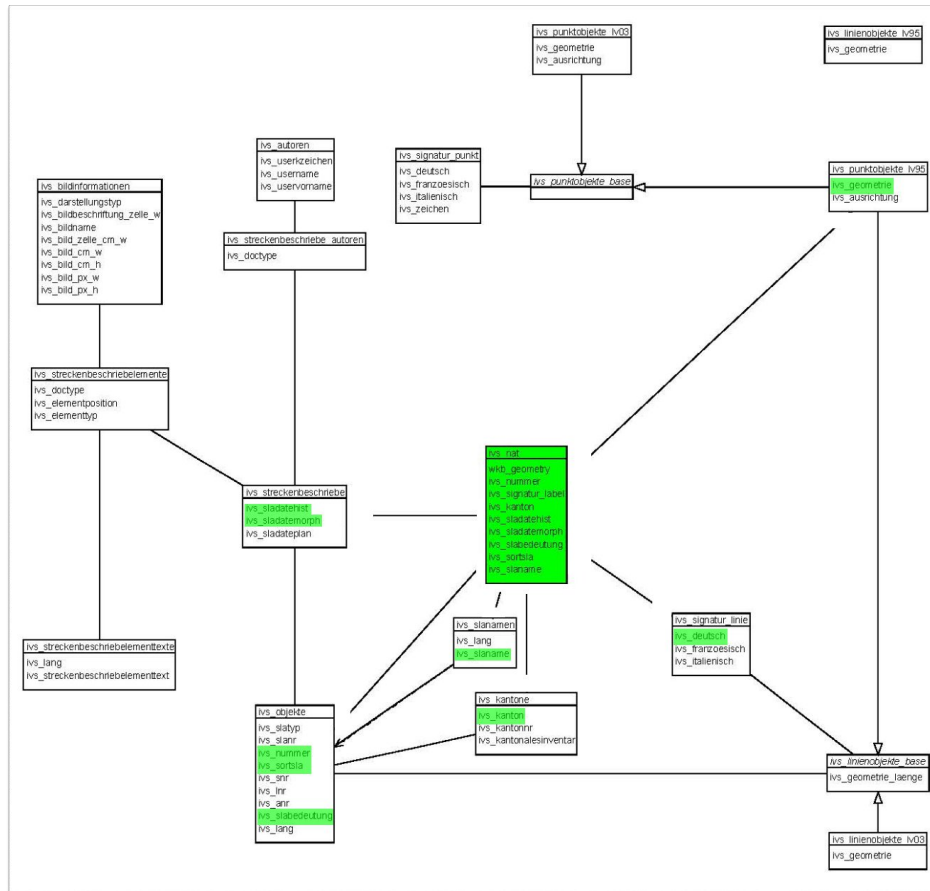


Figure 7: Diagramme de classe UML avec les classes et attributs mobilisés dans la VIEW

4.3.3. Exemple 3 : Zones réservées

Le 3^{ème} exemple est défini sur la base du MGDM Zones réservées (76.1⁵¹) et du rapport de mise en œuvre⁵². Le modèle StandardSymbology⁵³ qui est exploré à la section 5 étant défini en INTERLIS 2.4, il est néanmoins nécessaire d'adapter les modèles de données pour qu'ils puissent être compatibles.

La définition de la VIEW mobilise 2 différentes classes au travers de la combinaison de la vue JOIN OF et du recours à la clause WHERE.

```
INTERLIS 2.4;  
MODEL Planungszone_V2_d_A  
AT "mailto:maxime.collombin@heig-vd.ch"  
VERSION "2023-12-04" =  
    IMPORTS Planungszone_V2;  
    TOPIC PZ_d EXTENDS Planungszone_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 Planungszone V2.Geobasisdaten.Planungszone, Planungszone 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 Planungszone V2_d_A.

```

Bloc de code 9: premier modèle dérivé⁵⁴ du modèle Planungszone V2 avec la définition de la VIEW

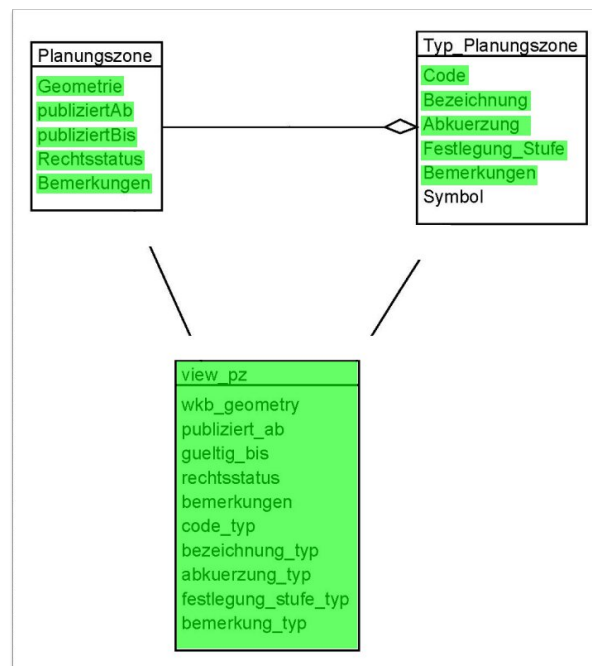


Figure 8: Diagramme de classe UML avec les classes et attributs mobilisés dans la VIEW

⁵⁴ https://github.com/MediaComem/FGDM4GS/blob/main/WP2-WP3/model/Planungszone V1_1/Planungszone V2_d_A.ili

4.3.4. Exemple 4 : Plan sectoriel des transports Partie infrastructure route

Le 4^{ème} exemple est défini sur la base du MGDМ Plan sectoriel des transports Partie infrastructure route (72.1⁵⁵). Le modèle Standard Symbology⁵⁶ qui est exploré à la section 5 étant défini en INTERLIS 2.4, il est néanmoins nécessaire d'adapter les modèles de données pour qu'ils puissent être compatibles. Les modèles importés dans le modèle étant défini dans la version 2.3 d'INTERLIS, il n'a pas été possible de définir le modèle dérivé sans rencontrer des erreurs, ce pourquoi il a été préconisé de définir le modèle dans la version 2.3.

Un attribut *Description* est défini sous forme d'un élément *STRUCTURE MultilingualMText* (Structure imbriquée = un attribut par langue). Ceci implique que cet attribut *Description* pourrait être traduit, soit par un seul attribut de cet élément STRUCTURE, soit par la concaténation de plusieurs attributs qui représentent chacun une traduction.

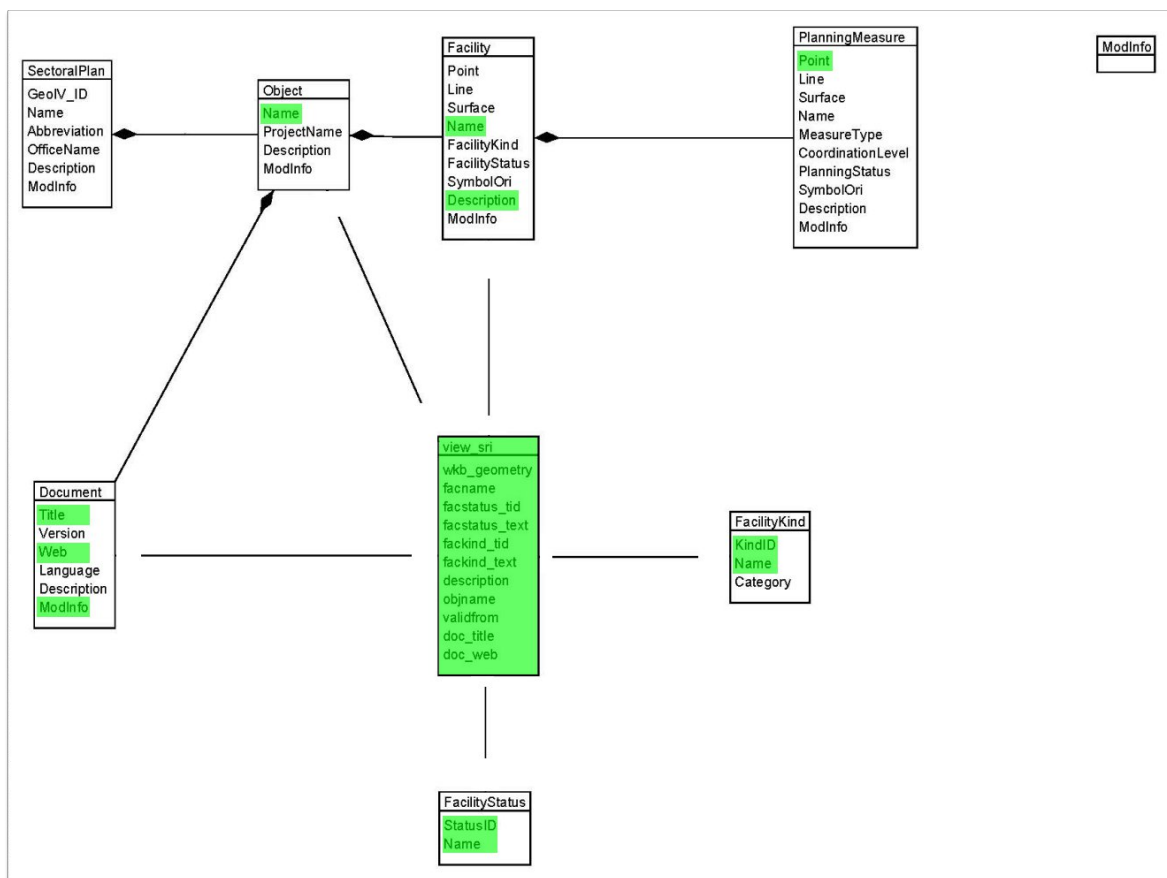


Figure 9: Diagramme de classe UML avec les classes et attributs mobilisés dans la VIEW

⁵⁵ <https://geobasisdaten.ch/?search=72.1>

⁵⁶ <https://models.interlis.ch/refhb24/StandardSymbology.ili>

La définition de la VIEW mobilise 10 différentes classes au travers de la combinaison de la vue JOIN OF et du recours à la clause WHERE avec plusieurs conditions.

```
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.
    Catalogue_FacilityStatus.FacilityStatus,
    BaseModel_SectoralPlans_Catalogues_V1_4.
    Catalogue_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; other option for the geometry */
    /* 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.
```

Bloc de code 10: modèle dérivé⁵⁷ du modèle BaseModel_SectoralPlans_LV95_V1_4 avec la définition de la VIEW

4.4. Recommandations

4.4.1. Création de modèle dérivé

Afin de ne pas altérer les modèles de base, nous suggérons la création de modèles dérivés pour la définition de VIEWS à des fins de publication de géoservices. Cela permet de garder une séparation des responsabilités, d'un côté purement sur la description du modèle, de l'autre les éléments de publication.

4.4.2. Vue de type projection (PROJECTION OF)

Nous suggérons l'utilisation de la vue projection (mot-clé PROJECTION OF) pour des modèles simples ou il s'agit de diffuser les données associés à une seule classe (par exemple avec renommage des intitulés des attributs).

4.4.3. Vue de type jonction (JOIN OF)

La vue de type jonction (mot-clé JOIN OF) associée à la clause WHERE permet d'associer les éléments de plusieurs CLASS en une vue. Ce type est à préférer pour des cas où des attributs de plusieurs CLASS sont réunis dans une vue.

4.4.4. Support ili2db du type projection et jonction

ili2db ne prend pas en charge les VIEW et ne permet donc pas de directement créer de vues à partir du modèle. Il serait donc important que ili2db prenne en charge les VIEWS pour faciliter la publication de géo-services. Selon précédents recommandations, les types projection et jonction pourraient être les seuls types (ou les types prioritaires) à implémenter.

4.4.5. Approche normative pour les vues

La publication de vues INTERLIS permet de définir une structure dénormalisée d'un modèle INTERLIS afin de faciliter la publication de géo-services d'une manière uniformisée pour un MGD. Il est proposé de mettre en route une approche normative, par exemple :

- avec une nouvelle norme eCH
- avec une adaptation des normes existantes : eCH-0031 et eCH-0056

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

5. Modèle de représentation

Depuis juin 2014, dans le cadre des travaux concrets d'harmonisation des géodonnées de base relevant du droit fédéral, notamment pour l'élaboration de modèles de géodonnées minimaux (MGDM), des recommandations soutiennent les communautés d'information spécialisées (FIG) dans l'élaboration de modèles de représentation (DstM).

La document de ces recommandations précise que « *les services spécialisés compétents de la Confédération doivent notamment s'appuyer sur ces recommandations lors de la définition de DstM pour les modèles de géodonnées minimaux (MGDM) afin d'obtenir un type de description de représentation uniforme. Cela simplifie la mise en œuvre ultérieure dans les applications.* », qu'il s'agit donc « *en premier lieu de représentation uniforme qui doit être mis à disposition au moyen du service de représentation* » et que « *les portails web qui seront mis en place dans le cadre de l'INDG doivent également pouvoir utiliser ces DstM* ».

L'approche prescrite consiste à documenter les informations de représentation graphique sous différentes formes possibles (voir figure 13) : sur la base d'un modèle tabulaire, d'un document PDF voire d'une description au format OGC SLD/SE. Malgré cette diversité des possibles, on peut constater aujourd'hui que ces recommandations sont très peu suivies avec seulement ~3% des entrées sur *geobasisdaten.ch* accompagnées d'une telle documentation. Nous émettons donc les hypothèses suivantes :

- le fait de se baser sur des définitions de styles selon un modèle orienté objet peut créer une confusion et potentiellement même amener à des conflits - p.ex pour le cas où des styles ont été définis pour deux CLASS, mais le service web à implémenter doit combiner les éléments des deux CLASS de façon aplatie. La mise à disposition d'une vue qui représente une structure aplatie permettra cela plus facilement.
- le fait de décrire les informations de représentation graphique en INTERLIS pourrait permettre de publier les modèles de représentation graphique avec les modèles conceptuels sur les dépôts de modèles et ainsi favoriser le processus de mise en oeuvre des géodonnées et géoservices basés sur les MGDM.

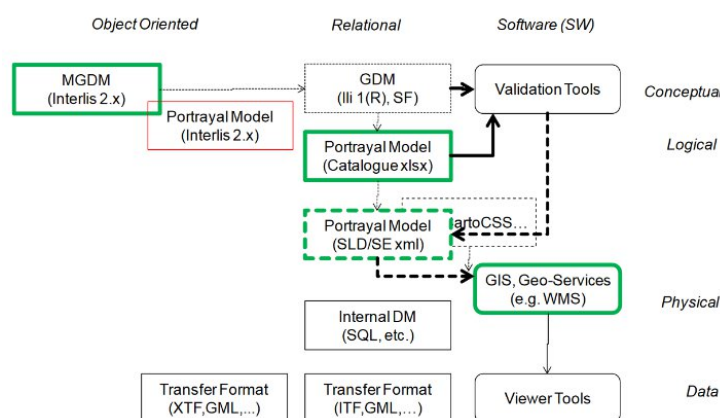


Figure 10: concept de représentation graphique croisé INTERLIS-OGC

INTERLIS définit un modèle graphique (.ili) et des signatures graphiques (.xtf). La signature graphique (.xtf) est liée au modèle graphique (.ili) lui-même lié au modèle de données. L'association du modèle graphique, des signatures graphiques et des données permettent le rendu cartographique (voir Figure 11: comparaison entre les concepts de représentation graphique INTERLIS (à gauche) et de l'OGC (à droite)) .

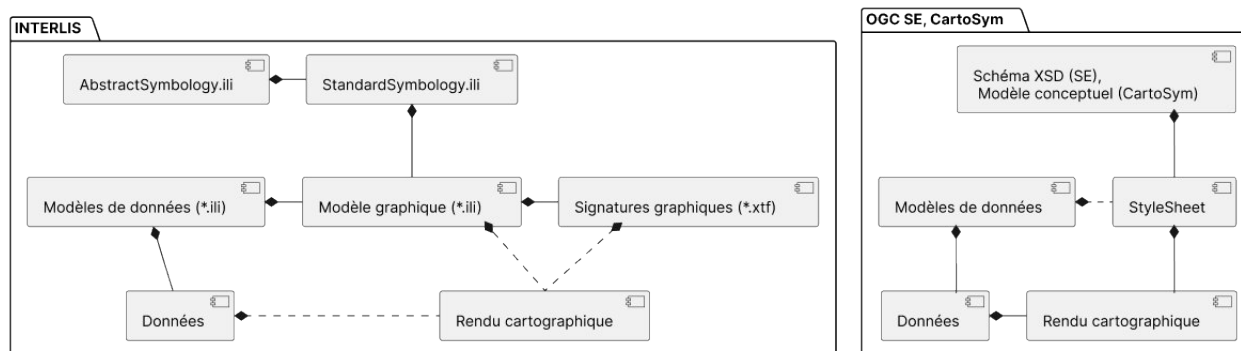


Figure 11: comparaison entre les concepts de représentation graphique INTERLIS (à gauche) et de l'OGC (à droite)

Du point de vue de l'OGC, les standards OGC Symbology Encoding 1.1.0⁵⁸ et OGC Styled Layer Descriptor 1.1.0⁵⁹ préconisés par les recommandations MGDM permettent la description de styles cartographiques, qui, une fois associés à des jeux de données ou des géoservices permettent de produire un rendu cartographique (voir Figure 11: comparaison entre les concepts de représentation graphique INTERLIS (à gauche) et de l'OGC (à droite)). Si ces standards sont modélisés par XML Schema et limités au seul encodage XML, l'OGC a publié en 2020 le standard OGC Symbology Conceptual Model : Core Part⁶⁰ décrivant un modèle conceptuel modulaire et extensible avec la possibilité de recourir à de multiples encodages pour favoriser une plus grande interopérabilité des styles cartographiques. L'évolution de ce dernier est en cours vers le futur standard OGC Cartographic Symbology (CartoSym) en venant compléter l'approche conceptuelle en proposant une approche multipartites avec 2 encodages de base : Cartographic Symbology in CSS (CartoSymCSS) et Cartographic Symbology in JSON (CartoSymJSON).

5.1. Méthodologie

Les capacités de représentation graphique d'INTERLIS sont analysées en regard des standards de l'Open Geospatial Consortium (OGC) et les MGDM sélectionnés sont mis en exemple, tout cela afin de déterminer dans quelle mesure la description graphique en INTERLIS pourrait être intégrée dans le processus de mise en oeuvre des MGDM.

Dans les sections suivantes, nous analysons les capacités de représentation graphique sur la base des classes d'exigences de CartoSym et proposons des exemples de signatures graphiques. Finalement des recommandations sont formulées en lien avec nos observations.

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. Comparaison INTERLIS et OGC

Selon la section 3.16 « Représentations graphiques » du manuel de référence INTERLIS, une représentation graphique se compose de définitions graphiques (« GraphicDef ») se basant toujours sur une vue ou une classe (mot-clé « BASED ON »). Au niveau conceptuel, une définition graphique a pour but d'affecter une signature graphique unique (symbole ponctuel, ligne, remplissage de surface, étiquette) à chaque objet de cette vue ou de cette classe se rapportant à la vue ou à la classe, par l'intermédiaire d'une ou de plusieurs règles de dessin (« DrawingRule ») et de générer ce faisant un ou plusieurs objets graphiques entraînant la représentation correspondante.

Les signatures graphiques uniques : « PolylineSign », « Symbolsign », « SurfaceSign », « TextSign » définis dans des fichiers de transfert correspondent respectivement à « Line-Symbolizer », « PointSymbolizer », « PolygonSymbolizer », « TextSymbolizer » dans le contexte d'OGC SE. Pour ce qui est de CartoSym, leurs propriétés sont définies dans les classes d'exigences « Basic Vector Features Styling », « Basic Labeling », « Advanced strokes » et « Advanced fills ».

La représentation Raster (« RasterSymbolizer » d'un « CoverageStyle » pour OGC SE ou classe d'exigence « Coverage Styling » pour CartoSym) n'est pas prise en charge par le modèle de symbologie INTERLIS.

Dans le contexte d'OGC SLD/SE, un « FeatureTypeStyle » est composé d'une ou de plusieurs règles (« Rule ») elles-mêmes composées de 0 à n filtres (« Filter ») et de 0 à n symboliseurs (« Symbolizer »). Les symboliseurs peuvent être de type Ligne (« LineSymbolizer »), Point (« PointSymbolizer »), Polygon (« PolygonSymbolizer ») ou Texte (« TextSymbolizer »).

Quant à CartoSym, la spécification est définie en classes d'exigences à savoir notamment Core⁶¹, Basic Vector Features Styling⁶², Basic Labelling⁶³, Advanced strokes⁶⁴, Advanced fills⁶⁵ et celles pour le Coverage Styling⁶⁶.

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

Table 3: classes d'exigences CartoSym et équivalents SE et INTERLIS

La section suivante effectue la comparaison entre INTERLIS OGC SLD/SE et CartoSym en tenant compte des classes d'exigences de CartoSym.

5.3. Classe d'exigences « Core »

Les exigences décrites dans cette section définissent la classe d'exigences de base de la symbologie. La comparaison des éléments de la classe d'exigence « Core » est décrit dans les sections suivantes.

5.3.1. Metadata

Selon CartoSym, les métadonnées contiennent des informations descriptives sur un « style » ou « stylesheet ». Elles peuvent comprendre un titre, un résumé, une description, une liste d'auteurs, une liste de mots-clés et une liste de classes de données géographiques. Tous les éléments de métadonnées sont préfixés par un point (.) comme premier caractère d'une ligne. Il n'en existe pas d'équivalent à proprement parler dans la syntaxe INTERLIS.

5.3.2. Styles

Selon CartoSym, un « style » ou « stylesheet » correspond au plus haut niveau hiérarchique du concept de représentation cartographique. Un style est composé de « StylingRule » et de métadonnées (« metadata »). Dans le contexte d'INTERLIS, un style équivaut à un « GRAPHIC ».

5.3.3. Styling rules

Selon CartoSym, une « StylingRule » spécifie qu'un Symbolizer particulier (définissant la manière de représenter les données) doit être appliqué si la règle est sélectionnée sur la base d'une expression « Selector » (par exemple, des conditions de propriétés d'entités ou des échelles de cartes). Une « StylingRule » peut contenir des règles imbriquées (nestedRules), qui ne seront appliquées que si le « Selector » de la règle parente est évalué à *true*, et dont les Symbolizers héritent du « Symbolizer » de la « StylingRule » parente et le remplacent.

La comparaison entre les règles de style (« StylingRule ») et les éléments du modèle de symbologie INTERLIS est décrit dans le tableau suivant.

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

Table 4: Styling Rules et équivalents SE et INTERLIS

Le nom d'une « StylingRule » est décrit par une expression suivie de « Of » à l'intérieur de l'élément « GRAPHIC » dans le modèle standard de symbologie. Si la règle n'est associée à aucun « Selector », on lui préfère alors l'expression par défaut « Symbol ».

Comme mentionné à la section précédente, le modèle standard de symbologie INTERLIS définit les « Symbolizer » suivants : « PolylineSign », « SymbolSign », « SurfaceSign » et « TextSign ». Autrement dit on peut considérer la notion OGC de « Symbolizer » et INTERLIS de « Sign(ature) » comme équivalente.

Le modèle de symbologie INTERLIS ne décrit pas d'opacité de manière globale pour un « Symbolizer ». Elle peut toutefois être spécifiée localement dans le contexte d'une couleur.

Aucun concept similaire à « nestedRules » n'a pu être identifié dans le modèle de symbologie INTERLIS.

5.3.4. Symbolizers

Selon CartoSym, un « Symbolizer » décrit comment représenter des données géographiques (par exemple, des entités vectorielles ou des données de couverture quadrillée) sur la base d'un ensemble de propriétés. Cette classe d'exigences ne définit que trois propriétés de base pertinentes pour la plupart des cas d'utilisation de la représentation. Des classes d'exigences distinctes et des parties ultérieures étendent cette classe avec des propriétés supplémentaires pour fournir des capacités de représentation plus avancées.

La comparaison entre les symboliseurs (« Symbolizers ») et les éléments du modèle de symbologie INTERLIS est décrit dans le tableau suivant.

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

Table 5: Symbolizers et équivalents SE et INTERLIS

5.3.5. Expressions

Selon CartoSym, les « expressions » sont des entités syntaxiques qui peuvent être évaluées pour déterminer leur valeur. Ce concept a une signification similaire à celle des expressions définies dans la norme OGC Filter Encoding 2.0⁶⁷ et peut également être mis en œuvre avec le Common Query Language (CQL2)⁶⁸ de l'OGC, sous réserve que les expressions définies ici ne soient pas limitées à l'évaluation d'une valeur booléenne, comme c'est le cas pour la première version de CQL2.

Les expressions peuvent être utilisées à deux endroits selon le modèle conceptuel Styles & Symbolologie : en tant que sélecteur définissant la condition d'application ou non d'une « StylingRule » (prédicat), ainsi que pour définir les valeurs des propriétés du Symbolizer.

Les expressions: « Identifier Expressions », « Literal Expressions » et « Operation Expressions » sont définies aux sections suivantes.

Les expressions d'instance (« Instance Expressions ») et de tableaux (« Array Expressions ») ne disposent pas d'équivalents ni pour SLD/SE ni pour le modèle de symbolologie INTERLIS.

5.3.6. Parameter Values

Selon CartoSym, l'utilisation d'une expression comme propriété du symboliseur est appelée « Parameter Value ». Une expression peut être substituée lorsqu'un type de données particulier est attendu, auquel cas l'expression sera résolue en fonction du type de données attendu. D'un point de vue conceptuel, toutes les propriétés du symboliseur peuvent être spécifiées à l'aide de n'importe quel type d'expression « Parameter Value », mais cela n'est pas requis par cette classe de conformité de base, et des classes d'exigences spécifiques telles que la classe d'exigences « Symbolizer Parameter Value Expressions » permettent d'adopter ce comportement. En l'absence de classe d'exigences spécifiant le contraire,

⁶⁷ <https://www.ogc.org/standard/filter/>

⁶⁸ <https://www.ogc.org/standard/cql2/>

seules les expressions littérales compatibles avec le type de données attendu sont autorisées comme valeurs de paramètres.

Le concept de « Parameter Values » n'est pas possible dans le modèle StandardSymbology.

5.3.7. Identifier Expressions

Selon CartoSym, une « Identifier Expressions » est représentée par une propriété textuelle qui est résolue au moment de l'exécution.

La comparaison entre les expressions d'identifiants (« Identifier Expressions ») et les éléments du modèle de symbologie INTERLIS est décrit dans le tableau suivant.

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

Table 6: Identifier Expressions et équivalents SE et INTERLIS

Quasi aucune des capacités des « Identifier Expressions » n'est possible dans le modèle de symbologie INTERLIS si ce n'est « visualization.pass » qui correspond à l'attribut « Priority » se trouvant dans un « GRAPHIC » défini dans le modèle graphique.

5.3.8. Literal Expressions

Selon CartoSym, une « LiteralExpression » est une expression qui a une valeur fixe dans le codage du style. Un concept similaire de classe littérale a été défini à l'origine dans la section 7.5.1 de la norme OGC Filter Encoding 2.0.

La comparaison entre les expressions littérales (« Literal Expressions ») et les éléments du modèle de symbologie INTERLIS est décrit dans le tableau suivant.

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

Table 7: Literal Expressions et équivalents SE et INTERLIS

Les expressions littérales (« Literal Expressions ») sont toutes définies au niveau du type de données dans le modèle de données.

5.3.9. Operation Expressions

Selon CartoSym, une « Operation Expression » effectue une opération spécifiée par un opérateur sur des expressions. Cette classe d'exigences fondamentales ne définit que les opérateurs logiques et relationnels. Les implémentations doivent également prendre en charge les opérations de hiérarchisation. D'autres classes d'exigences introduisent la prise en charge d'opérateurs supplémentaires pour les opérations arithmétiques, textuelles et binaires. Sans la prise en charge de la classe d'exigences « Any right-hand operand », les opérateurs de droite des « RelationExpressions » sont limités aux « LiteralExpressions ».

La comparaison entre les expressions d'opérations (« Operation Expressions ») et les éléments du modèle de symbologie INTERLIS est décrit dans le tableau suivant.

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

Table 8: Operation Expressions et équivalents SE et INTERLIS

Les opérateurs relationnels (RelationalOperator) correspondent aux éléments : « == », « != », « <> », « <= », « >= », « < », « > » pouvant être associées à une clause WHERE dans l'élément « GRAPHIC » du modèle graphique INTERLIS.

Les opérateurs logiques (« LogicOperator ») correspondent aux éléments « AND », « OR », « NOT » pouvant être associées à une clause WHERE dans l'élément « GRAPHIC » du modèle graphique INTERLIS.

5.3.10. Default class member values

Les valeurs par défaut des membres de classes ne sont pas reprises car les paramètres associés interviennent déjà dans les classes précédemment définies.

5.4. Classe d'exigences « Basic Vector Features Styling »

Cette classe d'exigences ajoute un support pour définir comment remplir et rendre le trait d'une géométrie vectorielle. En outre, elle permet de spécifier un marqueur composé d'un ou de plusieurs graphiques, à rendre à l'emplacement exact d'une géométrie ponctuelle, au centroïde d'une géométrie polygonale ou à chaque point d'une géométrie linéaire.

La comparaison des éléments de la classe d'exigence « Basic Vector Features Styling » est décrit dans les sections suivantes.

5.4.1. Symbolizer extension for vector features

Les trois propriétés suivantes du Symbolizer sont définies par cette classe d'exigences :

- Fill
- Stroke
- Marker

La comparaison entre l'extension de symboliseur pour les entités vectorielles («Symbolizer extension for vector features») et les éléments du modèle de symbologie INTERLIS est décrit dans le tableau suivant.

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>

Table 9: Symbolizer extension for vector features et équivalents SE et INTERLIS

En comparaison avec l'extension du Symbolizer⁶⁹ pour les styles applicables aux données vectorielles de CartoSym, un remplissage (« Fill ») peut être décrit par l'élément <ili:Name>fill</ili:Name> à l'intérieur d'un « SurfaceSign » dans un modèle graphique INTERLIS.

En comparaison avec l'extension du Symbolizer pour les styles applicables aux données vectorielles de CartoSym, un trait (« Stroke ») peut être décrit par l'élément « Style » en référence à un type de ligne (« LineStyle_ ») à l'intérieur d'un « PolylineSign ».

5.4.2. Extended Identifiers

Cette classe d'exigences définit comme identifiants valides toutes les propriétés accessibles d'un élément représenté, le nom de ces propriétés étant le texte de l'identifiant. Ces identificateurs doivent être résolus avec la valeur de cette propriété.

La comparaison entre les identifiants étendus (« Extended Identifiers») et les éléments du modèle de symbologie INTERLIS est décrit dans le tableau suivant.

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

Table 10: Extended Identifiers et équivalents SE et INTERLIS

5.4.3. Colors

CartoSym définit les couleurs à l'aide des composantes rouge, verte et bleue. La possibilité de définir des couleurs dans d'autres espaces (par exemple, Lab, CMYK) est définie dans

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

des classes d'exigences distinctes et s'applique partout où cette classe de couleurs RVB de base est utilisée.

La comparaison pour les couleurs est décrit dans le tableau ci-dessous.

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

Table 11: Colors et équivalents SE et INTERLIS

Dans le modèle de symbologie INTERLIS, la couleur est définie selon l'espace nominatif LCh avec le paramètre additionnel «T» pour la transparence.

Au niveau de CartoSym, en revanche, la couleur est définie selon les paramètres rgb.

A des fins d'interopérabilité de styles cartographiques, nous recommandons de définir plusieurs espaces de couleurs dans le modèle de symbologie, qu'il s'agisse de CartoSym ou d'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>
```

Bloc de code 11: exemple de définition d'une couleur (ici blanc)

5.4.4. Fills

Selon CartoSym, un remplissage (ou « Fill ») définit les paramètres de symbolisation graphique nécessaires pour dessiner le remplissage d'une forme bidimensionnelle telle qu'un polygone. Des classes de conformité distinctes peuvent étendre cette classe avec des propriétés supplémentaires.

La comparaison pour le remplissage («Fills ») est décrit dans le tableau ci-dessous.

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>

Table 12: Fills et équivalents SE et INTERLIS

La classe SurfaceSign correspond à un PolygonSymbolizer dans SLD/SE. La couleur de remplissage est décrite par une référence («FillColor») à une couleur («Color») à l'intérieure d'un «SurfaceSign» dans une signature graphique INTERLIS.

La transparence («opacity») est définie par le paramètre «T» de la couleur «Color» à laquelle le remplissage («Fill») fait référence avec le paramètre «FillColor».

5.4.5. Strokes

Selon CartoSym, un «Stroke» définit les paramètres de symbolisation graphique pour dessiner un contour (par exemple, pour les géométries linéaires ou l'extérieur d'une géométrie polygonale).

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

Table 13: Stroke et équivalents SE et INTERLIS

La classe PolylineSign correspond à un LineSymbolizer dans SLD/SE et aux classes d'exigences 8. *Requirements Class "Basic Vector Features Styling"*⁷⁰ et 13. *Requirements Classes for Advanced strokes*⁷¹ dans CartoSym.

La signature graphique ci-dessous démontre comment représenter un «Strokes» en INTERLIS.

```
<?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:tld="1">
        <Name>black</Name>
        <L>0.0</L>
        <C>0.0</C>
        <H>0.0</H>
        <T>1.0</T>
      </Color>
      <!-- Polyline attributes -->
      <PolylineAttrs ili:tld="4001">
        <Width>0.01</Width>
      </PolylineAttrs>
      <!-- Line Styles -->
      <LineStyle_Solid ili:tld="21">
        <Name>Continuous</Name>
        <Color ili:ref="1"></Color>
        <LineAttrs ili:ref="4001"></LineAttrs>
      </LineStyle_Solid>
      <!-- Polyline Signs -->
      <PolylineSign ili:tld="3001">
        <ili:Name>continuous</ili:Name>
        <Style ili:ref="21">
          <PolylineSignLineStyleAssoc>
          </PolylineSignLineStyleAssoc>
        </Style>
      </PolylineSign>
    </StandardSigns>
  </ili:datasection>
</ili:transfer>

```

Bloc de code 12: exemple⁷² de définition d'un stroke en INTERLIS

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

5.4.6. Markers

Pour définir des implantations ponctuelles, INTERLIS définit la classe `SymbolSign` qui comporte les attributs `Name`, `Scale`, `Rotation`, `Color`, `Symbol` et `ClipSymbol` ainsi que les attributs `Geometry` et `Priority` (obligatoire) dans le modèle graphique. Le symbole est décrit dans la classe `FontSymbol` comme nous l'avons vu plus haut. Le paramètre `ClipSymbol` permet de définir un masque pour le symbole. Pour ce qui est des autres attributs, ils ont déjà été définis dans les autres signatures graphiques (`Sign`).

OGC Symbology Encoding (SE) définit le concept de `Mark` avec `MarkIndex` pour sélectionner un glyphe individuel dans un fichier de police (`Font`), ce qui correspond à un `FontSymbol` (INTERLIS avec UCS4). Notons que SE permet aussi une `Mark` définie par un `WellKnownName` (carré, cercle, triangle, étoile, croix et x) ou par un `ExternalGraphic` pouvant référencer une ressource image externe bitmap ou vector (ex. PNG, SVG).

CartoSym	SE	INTERLIS
marker	<PointSymbolizer>	<FontSymbol> with a <Geometry> or a reference to the glyph of a (<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

Table 14: Markers et équivalents SE et INTERLIS

Dans le modèle `StandardSymbology`, les symboles peuvent être définis soit relativement à une `Font` (glyphe), soit en décrivant leur géométrie dans la Signature graphique (fichier *.otf). Le recours à une ressource image externe (« `ExternalGraphic` ») n'est, en revanche, pas possible.

L'annexe 8.2 exemplifie une signature graphique pour un marqueur en INTERLIS.

5.4.7. Dot Graphics

Selon `CartoSym`, « `Dot Graphic` » est un type de graphique défini par cette classe d'exigences qui peut être utilisé comme élément d'un marqueur.

La comparaison de « `Dot Graphics` » est décrite dans le tableau suivant.

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 (<UCS4> parameter)
stroke	N/A	N/A

Table 15: Dot Graphics et équivalents SE et INTERLIS

Le modèle StandardSymbology offre 2 possibilités de représenter un « Dot Graphics ». La première consiste à décrire la géométrie du « Dot » via le paramètre « Geometry » à l'intérieur d'un « FontSymbol ». Cela n'est toutefois pas idéal compte tenu du nombre de coordonnées à faire intervenir. Il est donc préférable de recourir au glyphe d'une Font en utilisant « FontSymbol » et une référence « UCS4 ».

La signature graphique ci-dessous démontre comment représenter un « Dot Graphics » en INTERLIS.

```
<?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>black</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-->
      <!-- External symbols -->
      <!-- Tree -->
      <FontSymbol ili:tid="201">
        <Name>Dot</Name>
        <UCS4>0049</UCS4>
        <Font ili:ref="301"></Font>
      </FontSymbol>
    </StandardSigns>
  </ili:datasection>
</ili:transfer>
```

```

<!-- 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>

```

Bloc de code 13: exemple⁷³ de définition d'un dot en INTERLIS

5.4.8. Image Graphics

Selon CartoSym, une image est un type de graphique défini par cette classe d'exigences qui peut être utilisé dans le cadre d'un marqueur.

La comparaison de « Image Graphics» est décrit dans le tableau suivant.

CartoSym	SE	INTERLIS
image	<Graphic>	<UCS4> & 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

Table 16: Image Graphics et équivalents SE et INTERLIS

La seule possibilité permettant de représenter une images dans une signature graphique INTERLIS et de faire référence à une image vectorielles préalablement intégrée dans une « Font » et l'utiliser dans un « FontSymbol ». La logique de l'exemple est la même que pour « Dot Graphics ».

5.4.9. Text Graphics

Selon CartoSym, un texte est un type de graphique défini par cette classe d'exigences qui peut être utilisé dans le cadre d'un marqueur.

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

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

Table 17: Text Graphics et équivalents SE et INTERLIS

Les explications ainsi qu'un exemple de signature graphique sont définies à la section suivante en lien avec l'étiquetage (Labeling).

5.5. Classe d'exigences « Labeling »

5.5.1. Basic Labeling & Font Outlines

Cette classe d'exigences prend en charge les étiquettes placées « au-dessus » des entités géographiques, moyennant éventuellement un algorithme de placement. Comme les marqueurs, les étiquettes héritent de la classe MultiGraphic et peuvent contenir plusieurs éléments graphiques.

La comparaison de « Basic Labeling & Font Outlines » est décrit dans le tableau suivant.

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	 with <SvgParameter name="font-family">, <SvgParameter name="font-size">, <SvgParameter name="font-style">, <SvgParameter name="font-weight">	 with <Name>, <Internal>, <Type> & <BottomBase>
FontOutline	<Halo>	N/A

Table 18: Basic Labeling et Font Outlines et équivalents SE et INTERLIS

Pour définir des implantations de type texte, INTERLIS définit la classe « TextSign » qui comporte les attributs « Name », « Height », « Weight », « Slanted », « Underlined », « Striked », « ClipBox », « Font » et « Color » ainsi que les attributs « Geometry », « Rotation » et « Priority » (obligatoire) dans le modèle graphique. Les paramètres « Height », « Weight », « Slanted », « Underlined » et « Striked » permettent de définir la taille, le poids, l'inclinaison, le soulignement et le barré du texte. Le paramètre « ClipBox » permet de définir une boîte de délimitation pour le texte. Pour ce qui est des autres attributs, ils ont déjà été définis dans les autres signatures graphiques (« Sign »). Le recours à l'attribut « Spacing » dé-

fini dans la classe « FontSymbol » ne semble pas possible alors qu'il semble dédié avec des symboles de type texte tel que défini dans le modèle StandardSymbology (cf remarque “*!! only for text symbols (characters)*”). Une association semble manquer dans le modèle entre « TextSign » et « FontSymbol ».

La signature graphique ci-dessous démontre comment représenter un « Basic Labeling » en INTERLIS.

```
<?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_Graphics</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>black</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>
      </TextSign>
    </StandardSigns>
  </ili:datasection>
</ili:transfer>
```

```

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

```

Bloc de code 14: exemple⁷⁴ de définition d'un label en INTERLIS

5.6. Classe d'exigences « Advanced strokes »

Les capacités de la classe d'exigence « Advanced strokes » sont décrites dans les sections suivantes.

5.6.1. Dashes

Cette classe d'exigences ajoute des capacités pour les traits avec la possibilité de définir le motif des tirets et des espaces utilisés pour peindre le trait, ainsi que le décalage sur le rendu du tableau de tirets associé.

La comparaison des traitillés (« Dashes ») est décrit dans le tableau suivant.

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

Table 19: Dashes et équivalents SE et INTERLIS

Un « dashPattern » peut être défini par l'attribut « DLength » de la structure « DashRec » étendue par la classe « LineStyle_Dashed ».

Un « dashOffset » peut être défini dans un « PolylineSign » par le paramètre « Offset » de l'association « PolylineSignLineStyleAssoc ».

La signature graphique ci-dessous démontre comment représenter un « dashPattern » et un « dashOffset » en INTERLIS.

```

<?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:tld="1">
      <Name>black</Name>
      <L>0.0</L>
      <C>0.0</C>
      <H>0.0</H>
      <T>1.0</T>
    </Color>
    <!-- Polyline attributes -->
    <PolylineAttrs ili:tld="4001">
      <Width>0.01</Width>
      <!-- Join type -->
      <!-- Other possible values are: round, bevel -->
      <Join>miter</Join>
      <!-- Note: MiterLimit is only available for miter join type -->
      <MiterLimit>10.0</MiterLimit>
      <!-- Cap type -->
      <!-- Other possible value: round -->
      <Caps>butt</Caps>
    </PolylineAttrs>
    <!-- Line Styles -->
    <!-- Dashed -->
    <LineStyle_Dashed ili:tld="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>
  </StandardSigns>
</ili:datasection>

```

```

<!-- 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>

```

Bloc de code 15: exemple⁷⁵ de définition d'un dashPattern et d'un dashOffset en INTERLIS

5.6.2. « Casing » et « Centerline »

Cette classe d'exigences définit la prise en charge des « Casing and Centerline ».

La comparaison des enveloppes et ligne médiane (« Casing and Centerline ») est décrit dans le tableau suivant.

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

Table 20: Casing and Centerline et équivalents SE et INTERLIS

Un « casing » et un « center » peuvent être définis par une superposition de « StandardSigns » contenant des « LineStyle_Solid » ou « LineStyle_Dashed ».

La signature graphique ci-dessous démontre comment représenter un casing et un centerline en INTERLIS.

```

<?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/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 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:tld="1">
      <Name>black</Name>
      <L>0.0</L>
      <C>0.0</C>
      <H>0.0</H>
      <T>1.0</T>
    </Color>
    <PolylineAttrs ili:tld="10">
      <Width>5</Width>
      <Join>miter</Join>
      <Caps>butt</Caps>
    </PolylineAttrs>
    <LineStyle_Solid ili:tld="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:tld="2">
      <Name>white</Name>
      <L>100.0</L>
      <C>0.0</C>
      <H>0.0</H>
      <T>1.0</T>
    </Color>
    <PolylineAttrs ili:tld="11">
      <Width>3</Width>
      <Join>miter</Join>
      <Caps>butt</Caps>
    </PolylineAttrs>
    <LineStyle_Solid ili:tld="21">
      <Name>Centerline</Name>
      <Color ili:ref="2"></Color>
      <LineAttrs ili:ref="11"></LineAttrs>
    </LineStyle_Solid>
  </StandardSigns>
</ili:datasection>
</ili:transfer>

```

Bloc de code 16: exemple⁷⁶ de définition d'un casing et center en INTERLIS

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

5.7. Classe d'exigences « Advanced fills »

Les capacités de la classe d'exigences « Advanced fills » sont décrits dans les sections suivantes.

5.7.1. « Hatch fills »

Cette classe d'exigences permet de définir des remplissages de hachures sous la forme d'un motif de lignes contrôlées par :

- la largeur (épaisseur) de la ligne,
- la distance entre deux lignes,
- l'angle des lignes.

La comparaison du remplissage par hachure est décrit dans le tableau suivant .

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>

Table 21: Hatch fills et équivalents SE et INTERLIS

Comme indiqué dans le tableau ci-dessus, l'attribut « HatchSymb » de l'association « SurfaceSignHatchSymbAssoc » permet de représenter des « Hatch fills ». Il doit faire référence à un « PolylineSign » dans une classe « SurfaceSign ». Il est à noter que cette capacité de rendu n'existe pas avec SLD/SE mais parfois par extension logicielle (VendorOption de GeoServer) du standard.

La signature graphique ci-dessous représente un exemple de « HatchStyle » en INTERLIS.

```
<?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:tld="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>black</Name>
    <L>0.0</L>
    <C>0.0</C>
    <H>0.0</H>
    <T>1.0</T>
  </Color>
  <!-- Polyline attributes -->
  <PolylineAttrsiili:tid="4001">
    <Width>0.01</Width>
    <Join>round</Join>
    <Caps>butt</Caps>
  </PolylineAttrsi>
  <!-- Line Styles -->
  <LineStyle_Solidili:tid="21">
    <Name>LineSolid_01</Name>
    <Colorili:ref="2"></Color>
    <LineAttrsiili:ref="4001"></LineAttrsi>
  </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"></HatchSymb>
    <!-- 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>

```

Bloc de code 17: exemple⁷⁷ de définition d'un HatchStyle en INTERLIS

5.7.2. « Dot Pattern fills »

Cette classe d'exigences étend les capacités de remplissage (« Fill ») en ajoutant la prise en charge des remplissages de points avec un motif basé sur des points.

5.7.3. « Stipple fills »

Cette classe d'exigences définit les remplissages en pointillé. Un remplissage en pointillé est un motif de points défini par une densité.

La comparaison du remplissage en pointillé (« StippleStyle ») est décrit dans le tableau suivant.

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

Table 22: Stipple fills et équivalents SE et INTERLIS

Comme indiqué dans le tableau ci-dessus, il n'est pas possible de représenter de remplissage en pointillé (« StippleStyle ») avec le modèle actuel de symbologie INTERLIS.

Le modèle StandardSymbology ne semble pas non plus prendre en charge le remplissage avec un « Pattern » (motif). Des questions restent également ouvertes telles que l'utilisation de l'attribut Clip (inside, outside) du modèle StandardSymbology. Le paramètre HatchOrg ne semble pas pouvoir être utilisé correctement non plus.

5.8. Classe d'exigence pour les expressions additionnelles

5.8.1. Classe d'exigences « Symbolizer Parameter Value Expressions »

Selon CartoSym, cette classe d'exigences permet d'utiliser des ParameterValues, c'est-à-dire des expressions arbitraires (pas seulement des littéraux), pour toutes les propriétés du Symbolizer. L'exemple caractéristique d'utilisation de ce concept est une représentation en symboles proportionnels. Ce concept n'étant pas possible dans la syntaxe INTERLIS, cette classe d'exigence n'est pas implémentée.

5.8.2. Classe d'exigences « Any right-hand operands »

Selon CartoSym, cette classe d'exigences supprime la restriction selon laquelle les opérandes du côté droit des RelationExpressions sont limités aux LiteralExpressions. Pour les expressions codées en CQL2, elle est analogue à la classe d'exigences CQL2 relative à la

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

comparaison propriété-propriété. Les recours aux expressions sont limitées dans la syntaxe INTERLIS et cette classe d'exigence n'est pas implémentée.

5.8.3. Classe d'exigences « Conditional Expressions »

CartoSym définit des expressions conditionnelles (« condition », « thenExp » et « elseExp ») de manière à pouvoir appliquer une « StylingRule » si les critères définis sont remplis. Aucun analogue n'existe dans la syntaxe INTERLIS.

5.8.4. Classe d'exigences « Variables »

Selon CartoSym, cette classe d'exigences ajoute un support pour la définition de variables qui peuvent être utilisées pour faciliter la réutilisation des définitions dans les encodages, ainsi que pour fournir des éléments configurables qui peuvent être associés à un contrôle d'application tel qu'un curseur ou un contrôle déroulant. Aucun analogue n'existe dans la syntaxe INTERLIS.

5.8.5. Classe d'exigences « Arithmetic Operators »

Selon CartoSym, cette classe d'exigences ajoute la prise en charge des opérations arithmétiques. Aucun analogue n'existe dans la syntaxe INTERLIS.

5.8.6. Classe d'exigences « Text Relation Operators »

Selon CartoSym, cette classe d'exigences ajoute un support plus avancé pour l'évaluation de la relation entre les valeurs de texte, y compris l'opérateur « like », « starts with », « contains », « ends with », ainsi que des variantes « not » pour tous ces éléments. Aucun analogue n'existe dans la syntaxe INTERLIS.

5.8.7. Classe d'exigences « Math Functions »

Selon CartoSym, cette classe d'exigence ajoute un support pour les fonctions mathématiques qui ne sont toutefois pas prise en compte par INTERLIS ce qui pourrait nécessiter une extension du modèle interne.

5.8.8. Classe d'exigences « Array Relation Functions »

Selon CartoSym, cette classe d'exigence ajoute un support pour la manipulation des « arrays ». Aucun analogue n'existe dans la syntaxe INTERLIS ce qui pourrait nécessiter une extension du modèle.

5.8.9. Classe d'exigences « Text Manipulation Functions »

Selon CartoSym, cette classe d'exigence ajoute un support pour la manipulation de textes. Aucun analogue n'existe dans la syntaxe INTERLIS ce qui pourrait nécessiter une extension du modèle.

5.9. Exemples sur la base des modèles sélectionnés

5.9.1. Exemple 1 : Axes des routes nationales

Le 1^{er} exemple est défini sur la base du MGDM Axes des routes nationales (86.1). Le modèle de représentation et les signatures graphiques qui lui sont associées sont définies sur la base de la VIEW de la section 4: Axes des routes nationales à laquelle l'attribut «Type» a expressément été ajouté pour servir la représentation graphique. D'après la documentation du modèle⁷⁸, le graphique doit être défini sur la base de la valeur de l'attribut «AxisType» intervenant dans un catalogue d'objet accessible sur le dépôt de modèles de la confédération⁷⁹.

La représentation du MGDM⁸⁰ mobilise 4 différents objets:

- L'objet «segment d'axe» correspondant à un « PolylineSign »
- L'objet «secteur» (Point de repère) correspondant à un « SymbolSign »
- L'objet « point de calage » correspondant à un « SymbolSign »
- L'objet « matérialisation physique » correspondant à un « SymbolSign »



Figure 12: rendu cartographique du MGDM Axes des routes nationales

Problèmes:

1. La définition du modèle graphique pour l'objet «segment d'axe» n'est pas possible car elle se base sur les valeurs des TID de la classe « AxisCatalogs_V1_1.AxisCatalogs.AxisType » du catalogue AxisCatalogs_V1_1 cité plus haut qui ne sont pas supportés par la « DrawingRule » (erreur: `.\Axis_V1_1_gm.ili:15:expecting NAME, found '89'`). Cette erreur relève toutefois uniquement du catalogue d'objets qui pourrait être mis à jour pour respecter les critères imposés par la « DrawingRule ».

⁷⁸ <https://www.astra.admin.ch/astra/fr/home/services/weitere-bereiche/geoinformation/geodonnees-de-base/reseau-des-routes-nationales.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/description_du_modeledegeodonneesminimalreseaudesroutesnationales.pdf

2. La définition du modèle graphique pour l'objet «secteur» (Point de repère) n'est pas non plus possible car elle nécessite le recours aux valeurs de l'attribut « Sequence » de la classe « Sector » ce qui n'est actuellement pas possible en INTERLIS (cf section observations pour la Classe d'exigences Symbolizer Parameter Value Expressions)
3. La définition des modèles graphiques pour les objets « point de calage » et « matérialisation physique » n'est pas possible car aucun détail ne figure dans la documentation du modèle permettant de déterminer avec certitude à quelle classes et attributs ils se rapportent.

La base de la définition du modèle graphique est fournie ci-dessous.

```
INTERLIS 2.3;
!! Version | Date          | Who      | Modification
!!-----
!! 1.0      | 2024-01-04 | HEIG-VD | - Création du modèle
MODEL Axis_V1_1_gm
AT "mailto:maxime.collombin@heig-vd.ch"
  VERSION "2024-01-04" =
  IMPORTS Axis_LV95_V1_1_d;
  !! L'import du modèle Axis_LV95_V1_1 est également nécessaire vis à vis de la classe Sector
  IMPORTS Axis_LV95_V1_1;
  IMPORTS StandardSymbology;
  !! SIGN BASKET StandardSymbology ~ StandardSymbology.StandardSigns
  !! !! Définition des objets «segment d'axe» (PolylineSign) sur la base de la classe AxisCatalogs_V1_1.AxisCatalogs.AxisType du catalogue AxisCatalogs_V1_1
  !! OBJECTS OF PolylineSign: 89aa138a-999a-554e-b120-13721dbbb055, 1f22835b-e726-bd40-b95d-5b5de18db20a, bf44d30a-b831-c242-beb3-1b252d7be688 ;
  !! !! Les TID ne sont pas supportés par la DrawingRule (erreur: .\Axis_V1_1_gm.ili:15:expecting NAME, found '89')
  !! !! Définition des objets «secteur» (Point de repère) (SymbolSign)
  !! OBJECTS OF SymbolSign: ;
  !! !! La définition de cette DrawingRule n'est pas possible car elle nécessite de recourir à des ParameterValues
  !! Définition du TOPIC pour les GRAPHIC ( ~ styleSheets)
  TOPIC Graphics =
    DEPENDS ON Axis_LV95_V1_1_d.Axis_d;
    DEPENDS ON Axis_LV95_V1_1.Axis;

    !! Définition du GRAPHIC (styleSheet) pour les segments d'axe
    GRAPHIC AxeSegment
      BASED ON Axis_LV95_V1_1_d.Axis_d.view_Axis =
        !! Définitions des DrawingRules (stylingRules)
    END AxeSegment;

    !! Définition du GRAPHIC (styleSheet) pour les points de repère
```

```

!! Définitions impossibles en raison des TID non supportées
GRAPHIC Sector
  BASED ON Axis_LV95_V1_1.Axis.Sector =
    !! Définitions des DrawingRules (stylingRules)
    !! Définitions impossibles car nécessitant l'accès aux valeurs de l'attribut Sequence de la
classe Sector (ParameterValue)
  END Sector;

!! Définition du GRAPHIC (styleSheet) «point de calage» (impossible faute d'informations
disponibles)
!! GRAPHIC Kalibrierungspunkt
!!   BASED ON <classe & attribut non spécifiés> =
!! END Kalibrierungspunkt;
!! Définition du GRAPHIC (styleSheet) «matérialisation physique» (impossible faute d'infor-
mations disponibles)
!! GRAPHIC PhysikalischeVersicherung
!!   BASED ON <classe & attribut non spécifiés> =
!! END PhysikalischeVersicherung;
END Graphics;
END Axis_V1_1_gm.

```

Bloc de code 18: modèle graphique⁸¹ du MGDM Axes des routes nationales

5.9.2. Exemple 2 : Inventaire des voies de communication historiques

Le 2^{ème} exemple est défini sur la base du MGDM Inventaire fédéral des voies de communication historiques (16.1, 17.1). Le modèle de représentation et les signatures graphiques qui lui sont associées sont définies sur la base de la VIEW de l'exemple 2 de la section 4 : Inventaire des voies de communication historiques. La représentation du MGDM⁸² mobilise 2 types d'implantations: linéaires (PolylineSign) basées sur la classe «ivs_linienobjekte_lv95» et ponctuelles (SymbolSign) basées sur la classe «ivs_punktobjekte_lv95».

Problèmes :

- La comparaison de valeurs littérales nécessitent généralement une chaîne de caractères ou le début et la fin sont bien définies (p.ex début et fin définies avec des guillemets). Les outils INTERLIS en place gèrent mal une chaîne de caractère composée de mots avec des espaces.
- Les SymbolSign nécessite le recours aux glyphes de la Font *ivs200440903* comme il est mentionné dans la documentation du modèle. Cette police n'est toutefois accessible nulle part.

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-historische-verkehrswege.pdf.download.pdf/description_du_modeledegeodonneesvoiesdecommunicationhistoriques.pdf

5.9.3. Exemple 3 : Zones réservées

Le 3ème exemple est défini sur la base du MGDm Zone réservé (76.1). Le modèle de représentation et les signatures graphiques qui lui sont associées sont définies sur la base de la VIEW de l'exemple 3bis de la section 4: Zones réservées. La « DrawingRule » du modèle graphique est définie sur la base de l'attribut: « festlegung_stufe_typ » («Disposition_Niveau_Type» en français). Les extraits de code ci-dessous correspondent au modèle et à la définition des signatures graphiques.

```
INTERLIS 2.4; !! V1.0      | 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 Graphics =
  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.
```

Bloc de code 19 : modèle graphique⁸³ du MGDm Zones Réservées

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 Planungszone 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:Planungszone V1_1_gm="http://www.interlis.ch/xtf/2.4/Planungszone V1_1_gm">
  <ili:headersection>
    <ili:models>
      <ili:model>StandardSymbology</ili:model>
      <ili:model>Planungszone V2_gm</ili:model>
    </ili:models>
    <ili:sender>HEIG-VD</ili:sender>
    <ili:comment>Planungszone V2 Grafic Modell Signatures</ili:comment>
  </ili:headersection>
  <ili:datasection>
    <!-- Define the StandardSigns -->
    <StandardSigns ili:bid="Planungszone Symb">
      <!-- Farbdefinition -->
      <Color ili:tId="KantonFarbe">
        <Name>C-Violet_moyen</Name>
        <L>81.69</L>
        <C>34.96</C>
        <H>308.4</H>
        <T>0.6</T>
      </Color>
      <Color ili:tId="GemeindeFarbe">
        <Name>C-Violet clair</Name>
        <L>88.57</L>
        <C>21.09</C>
        <H>306.48</H>
        <T>0.6</T>
      </Color>
      <Color ili:tId="AndereFarbe">
        <Name>C-Pourpre</Name>
        <L>94.86</L>
        <C>8.59</C>
        <H>297.89</H>
        <T>0.6</T>
      </Color>
      <Color ili:tId="BorderFarbe">
        <Name>C-Violet foncé</Name>
        <L>73.7</L>
        <C>37.41</C>
        <H>301.72</H>
        <T>1</T>
      </Color>
    </StandardSigns>
  </ili:datasection>
</ili:transfer>

```

```

</Color>
<!-- Define the 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>
<!-- Define the BorderWidth -->
<PolylineAttrs ili:tid="BorderWidth">
  <Width>2</Width>
</PolylineAttrs>
<!-- Define the BorderStyle -->
<LineStyle_Solid ili:tid="BorderStyle">
  <Name>BorderStyle</Name>
  <Color ili:ref="KantonFarbe"></Color>
  <LineAttrs ili:ref="BorderWidth"></LineAttrs>
</LineStyle_Solid>
<!-- Define the Border -->
<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>

```

Bloc de code 20: signatures graphiques⁸⁴ du MGDM Zones Réservées

5.9.4. Exemple 4 : Plan sectoriel des transports Partie infrastructure route

Le 4^{ème} exemple est défini sur la base du MGDM Plan sectoriel des transports Partie infrastructure route (72.1). Le modèle de représentation et les signatures graphiques qui lui sont associées sont définies sur la base de la VIEW de l'exemple 4 de la section 4: Plan sec-

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

toriel des transports Partie infrastructure route. D'après la documentation du modèle⁸⁵, les le graphiques doivent être définis sur la base des valeurs des attributs «FacilityStatus» et «PlanningStatus» intervenants dans un catalogue d'objet accessible sur le dépôt de modèles de la confédération⁸⁶.

Problème : Les valeurs de l'attribut «FacilityKind» de la classe «Facility» du modèle «BaseModel_SectoralPlans_Catalogues_V1_4» (au niveau du TOPIC «SectoralPlans_WithLatest-Modification») n'étant pas définis dans le catalogue « SectoralPlans_Catalogues_V1_4 », il n'est pas possible de définir le modèle et les signatures graphiques. Il ne s'agit cependant pas d'un problème lié au modèle StandardSymbology mais de la mise en oeuvre du MGDM.

5.10. Recommandations

5.10.1. Représentation raster

Le modèle StandardSymbology pourrait être mis à jour pour permettre la représentation Raster (« RasterSymbolizer » d'un « CoverageStyle » pour OGC SE ou classe d'exigence « Coverage Styling » pour CartoSym).

5.10.2. Espaces de couleurs

A des fins d'interopérabilité de styles cartographiques, nous recommandons de définir plusieurs espaces de couleurs dans le modèle de symbologie, qu'il s'agisse de CartoSym ou d'INTERLIS.

5.10.3. Formatage des expressions littérales

Les expressions sont limitées pour les sélecteurs dans le modèle graphique. Elles ne peuvent correspondre qu'à des valeurs numériques ou des strings basiques, sans espaces, ni accents, ni caractères spéciaux (voir par exemple les modèles du fichier IVS_V3_gm.ili⁸⁷ (WHERE ivs_signatur_label == #Nationale Bedeutung, historischer Verlauf mit viel Substanz). Il serait important de clarifier dans eCH-0031 le formatage des expressions littérales (ex. WHERE myAttribute == 'my space delimited value').

5.10.4. Transcodeur de style/symbologie

Pour favoriser l'interopérabilité entre INTERLIS et les autres langages de description de styles cartographiques, il s'agirait en premier lieu de créer un mapping entre INTERLIS, OGC SE & CartoSym puis implémenter des *parsers/writers* pour transcoder la définition INTERLIS en CartoSymJSON par exemple.

⁸⁵ <https://www.astra.admin.ch/astra/fr/home/services/weitere-bereiche/geoinformation/geodonnees-de-base/plan-sectoriel-des-transports--partie-infrastructure-routiere.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. Extension du modèle INTERLIS de description de symbologie

Les classes d'exigences suivantes ne sont pas ou que partiellement intégrées dans le modèle INTERLIS de description de symbologie

- Symbolizers (visibility)
- Identifier Expressions
- Image GraphicsStipple fills
- Classe d'exigence pour les expressions additionnelles
- Classe d'exigence pour les fonctions

Certaines exigences pourraient enrichir StandardSymbology, voire d'autres faire l'objet d'un modèle complémentaire AdvancedSymbology par exemple.

5.10.6. Intégration AbstractSymbology et StandardSymbology

Le modèle AbstractSymbology pourrait potentiellement être intégré dans le modèle StandardSymbology (comme d'ailleurs recommandé par M. Baertschi dans son travail de bachelior à l'ETHZ⁸⁸).

5.10.7. Mise à disposition des font dans un repository

Si la représentation graphique utilise des ressources particulières pour un graphique, il serait bien que les Font puissent être accessibles en ligne (p.ex. WESP Font) et que les ExternalGraphic puissent être également disponibles via une opération GetStyles ou alors il conviendrait préférablement de privilégier des ressources de type SVG.

5.10.8. Catalogue de styles

Il serait intéressant de publier des collections de styles afin de favoriser leur réutilisation ainsi que l'harmonisation des représentations cartographiques des géoservices publiés. Par exemple au travers de la spécification OGC API - Styles pourrait être envisagée à cet égard.

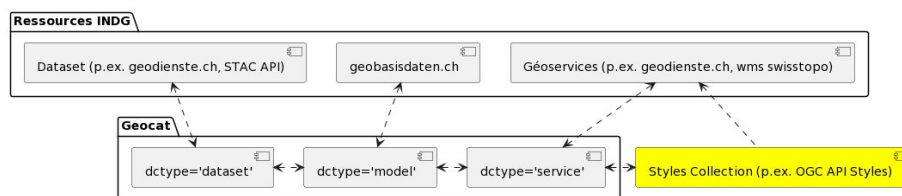


Figure 13: concept de mise en relation des ressources de l'INDG et de mise en oeuvre d'une collection de styles

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

5.10.9. Approche normative pour la symbologie

La mise à disposition de styles à l'aide d'INTERLIS peut avoir plusieurs avantages - comme p.ex : définition uniformisée des styles ; mise à disposition des styles au même endroit ; possibilité d'automatiser la création de styles en différents formats sur la même base. Il est proposé d'adopter une approche normative et de remplacer les recommandations de 2014 p.ex par une nouvelle norme eCH et d'adapter également les normes eCH-0031 et eCH-0056.

6. Automatisation de la création de géoservices

Considérant les limitations que les recommandations invitent à lever (notamment de l'outil ili2db) une automatisation ne peut être mise en œuvre. Toutefois, si on les considère levées, alors un tel processus serait composé des étapes suivantes :

- Etape 1 : comme montré dans le chapitre 4 et selon recommandations, il s'agit de créer une vue par la définition d'un modèle dérivé MGDM dédiée à la publication de géoservices
- Etape 2a : utilisé l'outil ili2db pour traduire les définitions du modèle de données INTERLIS dans un schéma de base de données (PostgreSQL/PostGIS, GeoPackage, ESRI FileGDB) et charger les données INTERLIS dans la base de données créée

nb. nécessite un support JOIN OF ... WHERE (selon recommandations) pour créer la vue correspondante dans le modèle physique de la base de données cible

- Etape 2b : transcoder la définition du modèle de représentation MGDM vers le format de description de style/symbologie prescrit par les bases normatives

nb. nécessite un outil de transcodage (selon recommandations) dont la conformance avec les standards OGC de modélisation de style/symbologie permet alors de favoriser l'interopérabilité des géoservices (ex. OGC SymCore/CartoSym, OGC API Styles)

- Etape 3 : produire et déployer dans l'infrastructure de publication les fichiers de configuration des géoservices (ex. pour mapserver, arcgis server, qgis-server, etc)

nb. selon des bases normatives comme eCH-0056 pour les services de téléchargement de données (ex. autour de l'OGC API Features), pour les services de consultation (ex. aujourd'hui autour de Basic WMS et Styleable WMS, demain des exigences similaires autour de l'OGC API Maps et Styles, etc)

7. Conclusion

Le projet FGDM4GS aura permis de mettre en évidence les possibilités et limites d'utiliser des VIEW ainsi que la modélisation de représentations avec des signatures graphiques afin de favoriser la publication de jeux de géodonnées et géoservices dénormalisés tout en laissant le modèle INTERLIS de base intouché.

Sur la base de quatre modèles INTERLIS sélectionnés, les modèles dérivés et les modèles de représentation ont été implémentés dans la mesure du possible. Néanmoins on peut constater un manque d'informations et de documentation. Les outils de compilation et de validation n'ont pas toujours aidé, et malgré de nombreux contacts pris avec les acteurs de la communauté INTERLIS pour combler ces difficultés, les exemples élaborés méritent encore réflexion. D'une certaine façon, cela confirme une sorte d'« immaturité » de l'environnement INTERLIS et des limites des pratiques sous-jacentes pour la symbologie. Cela participe aussi des recommandations formulées dans la section 5.10.

Bien que se heurtant actuellement à quelques limitations telles que relevées dans les recommandations, les prendre en compte pourrait constituer un véritable intérêt dans la cadre du processus de mise en œuvre des MGDm.

Il sera utile d'organiser un workshop dans la continuation de ce projet pour poursuivre les réflexions utiles avec des référents de la communauté INTERLIS et définir un plan d'actions. Pour déployer tout le potentiel et répondre à la problématique de ce projet, nous avons formulé les recommandations suivantes (voir chapitres 4.4 et 5.10) :

1. Création de modèle dérivé
2. Vue de type projection (PROJECTION OF)
3. Vue de type jonction (JOIN OF)
4. Support ili2db du type projection et jonction
5. Approche normative pour les vues
6. Représentation raster
7. Espaces de couleurs
8. Formatage des expressions littérales
9. Transcodeur de style/symbologie
10. Extension du modèle INTERLIS de description de symbologie
11. Intégration AbstractSymbology et StandardSymbology
12. Mise à disposition des font dans un repository
13. Catalogue de styles
14. Approche normative pour la symbologie

Dans la perspective d'un plan d'actions, il paraît découler de ces recommandations deux dimensions :

- dimension normative : nous proposons à la fois une approche normative pour la création de VIEW INTERLIS et également une approche normative pour l'extension de la symbologie INTERLIS existante : renouvellement des recommandations de 2014 sur les modèles de représentation et l'élaboration d'une nouvelle norme eCH et en parallèle une adaptation des normes eCH-0031 et eCH-0056
- dimension outillage (« tooling ») : nous proposons une adaptation des outils INTERLIS ili2c (compilateur) et ili2db pour tenir compte des VIEW INTERLIS (en priorité JOIN OF et PROJECTION OF). De plus, la possibilité de transcodage des styles INTERLIS vers les normes et standards d'interopérabilité en vigueur est à envisager.

8. Annexes

8.1. Procédure de sélection des modèles données pour cette étude

Description of the performed tasks :

- Creation of a python [script](#) to extract data from the [geobasisdaten.ch](#) API to an excel file (to keep only the necessary columns).
- Removal of entries that do not have an .ili data model
- Manual search of the relevant websites for links to the representation models (pdf, zip & excel)
- Use of linux script: "[check_xlsx.sh](#)" to test whether zips contain excel files
- Deletion of model entries that do not have excel files for representation models. We assume that these excel files represent the standardised description format and ignore the others.
- Use of the linux script "[getfilename.sh](#)" to print the names of excel files (for zip files containing several files, the names of these files are concatenated into a string of characters transcribed into an excel cell).
- Use of excel functions (=A2=A1; =AND(A1=TRUE,B1=TRUE)) to determine which models (data and representation) are listed several times, then aggregation of the related information and deletion of redundant entries.
- In order to select the models to be used, we have defined the following selection criteria: - entries in [geobasisdaten.ch](#) that have a link to a data model (.ili file) - entries from geobasisdaten.ch that have a link to a .xlsx representation model in Excel format (because these files are the closest to what could be considered standardised resources and the representation models are linked to the definition of views) - geobasisdaten.ch entries with a link to a geoservice or a usable data file - finally, the models are selected on the basis of their complexity

NOTE :

- The catalog of [geobasisdaten.ch](#) contains **346** entries (02.10.2023)
- **278/346** entries contain a link to a datamodel (.ili file)
- Why don't all the entries contain a link to a data model?
- **30/278** data models contain representation information
- Of these **30** models, **11** contain representation models in excel format.
- Of these **11** models **8** contain a link to a geoservice or a usable data file
- With regard to [geobasisdaten.ch](#), it might be appropriate to add a link to the representation models (for those that exist) in the API.
- For data models for which services exist on map.geo.admin.ch, it could be relevant to add a link to these services in the API.
- In the same way as in the definition of these services, the representation models used should be specified by including a link to the model or to the API endpoint.

Results

These steps allow to select the following models:

autorité	titre	id	Modèle minimal	Webpage	Représentation	Format	GetLegend Graphic	GetStyle	Complexité
ARE	Inventaire des logements et proportion de résidences secondaires	202.1	Lien	Lien	Lien	xlsx	Lien	Lien	-
ASTRA	Alignement des routes nationales	88.1	Lien	Lien	Lien	xlsx	Lien	Lien	-
ASTRA	Comptage de la circulation routière - réseau principal	13.1;14.1	Lien	Lien	Lien	xlsx	Lien	Lien	-
ASTRA	Inventaire fédéral des voies de communication historiques	16.1;17.1	Lien	Lien	Lien	xlsx	Lien	Lien	+
ASTRA	Plan sectoriel des transports Partie infrastructure route	72.1	Lien	Lien	Lien	xlsx	Lien	Lien	+
ASTRA	Réseau des routes principales	90.1	Lien	Lien	Lien	xlsx	Lien	Lien	-
ASTRA	Routes nationales	86.1	Lien	Lien	Lien	xlsx	Lien	Lien	+
ARE	Zones réservées	76.1	Lien	Lien	Lien	xlsx	Lien	Lien	+

Geocat

Geocat also allows to retrieve informations about MGDM.

One can for instance filter the records corresponding to models via the following [url](#). However, this does not provide links to related geoservices and representation models.

Another possibility is to send a request to the CSW service:

```
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&typeName=csw:Record&resultType=results&ElementSetName=full&maxRecords=3&lang=fr
```

but not the all the records correspond to MGDM and the filter on the dct:abstract LIKE 'Modèle de géodonnées minimal%' doesn't seem to work.

By using the id of the records obtained with the GetRecordById operation and the <http://www.opengis.net/cat/csw/2.0.2> schema, it is then possible to obtain the extended information linked to the record:

```
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
```

However, it is not possible to obtain the URLs of other linked records containing links to geoservices.

A final possibility lies in the Geonetwork API. It is possible to use the following endpoint:

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

and associate it with one of the following 2 JSON:

```
{
  "_source": {
    "includes": [
      "uuid",
      "resourceTitleObject.langfre",
      "resourceAbstractObject.langfre"
    ]
  },
  "from": 0,
  "size": 26,
  "query": {
    "bool": {
      "must": [
        {
          "query_string": {
            "query": "(any.langfre:(modèle de géodonnées minimal MGDm) OR any.common:(modèle de géodonnées minimal MGDm) OR resourceTitleObject.langfre:(modèle de géodonnées minimal MGDm)^2 OR resourceTitleObject.\\*:\\\"modèle de géodonnées minimal 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": {
      "must": [
        {
          "query_string": {
            "query": "(uuid:\\\"f3e3b347-cc58-4b29-bb87-a35eed4487ea\\\")",
            "default_operator": "AND"
          }
        }
      ]
    }
  },
  "track_total_hits": true,
  "sort": {"_id": "asc"}
}
```

but this doesn't allow the information to be linked either.

Conclusion

As will be explained later in the project documentation, in order to define the VIEWS and representation models in INTERLIS 2, it is necessary to cross-reference the information in the data models with the geoservices or associated data sets ("flat models") and the representation models (metadata, EXCEL or WMS GetStyle models), but neither Geocat nor geobasisdaten.ch currently allow this cross-referencing.

8.2. Définition d'un marker en INTERLIS

Exemple⁸⁹ de définition d'un marker en 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>black</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>Symbols</Name>
        <Internal>true</Internal>
      </Font>
    </StandardSigns>
  </ili:datasection>
</ili:transfer>
```

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>
  <Geometry>
    <FontSymbol_Surface>
      <Geometry>
        <geom:surface>
          <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>
  <Geometry>
    <FontSymbol_Polyline>
      <Color ili:ref="2"></Color>
      <Geometry>
        <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>
<!-- Circle -->
<FontSymbol ili:tid="102">
  <Name>Circle</Name>
  <!-- No FontSymbol_Surface is defined here, as it is not necessary for a circle
  and would also mean defining a large number of coordinates. -->
  <Geometry>
    <FontSymbol_Polyline>
      <Color ili:ref="2"></Color>
      <Geometry>
        <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>
<!-- Triangle -->
<FontSymbol ili:tid="103">
  <Name>Triangle</Name>
  <Geometry>
    <FontSymbol_Surface>
      <Geometry>
        <geom:surface>

```

```

        <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>
<Geometry>
    <FontSymbol_Polyline>
    <Color ili:ref="2"></Color>
    <Geometry>
    <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 -->
<!-- Note that this symbol could be made more complex -->
<FontSymbol ili:tld="104">

```

```

<Name>Star</Name>
<Geometry>
  <FontSymbol_Surface>
    <Geometry>
      <geom:surface>
        <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>
<Geometry>
  <FontSymbol_Polyline>
    <Color ili:ref="2"></Color>
    <Geometry>
      <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>
  <Geometry>
    <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:tld="105">
  <Name>Cross</Name>
  <Geometry>
    <FontSymbol_Polyline>
      <Color ili:ref="2"></Color>
      <Geometry>
        <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>
      <Geometry>

```

```

        <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 -->
<!-- Not necessary to define this symbol, as it can be obtained by rotating a cross -->
<!-- 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>

```