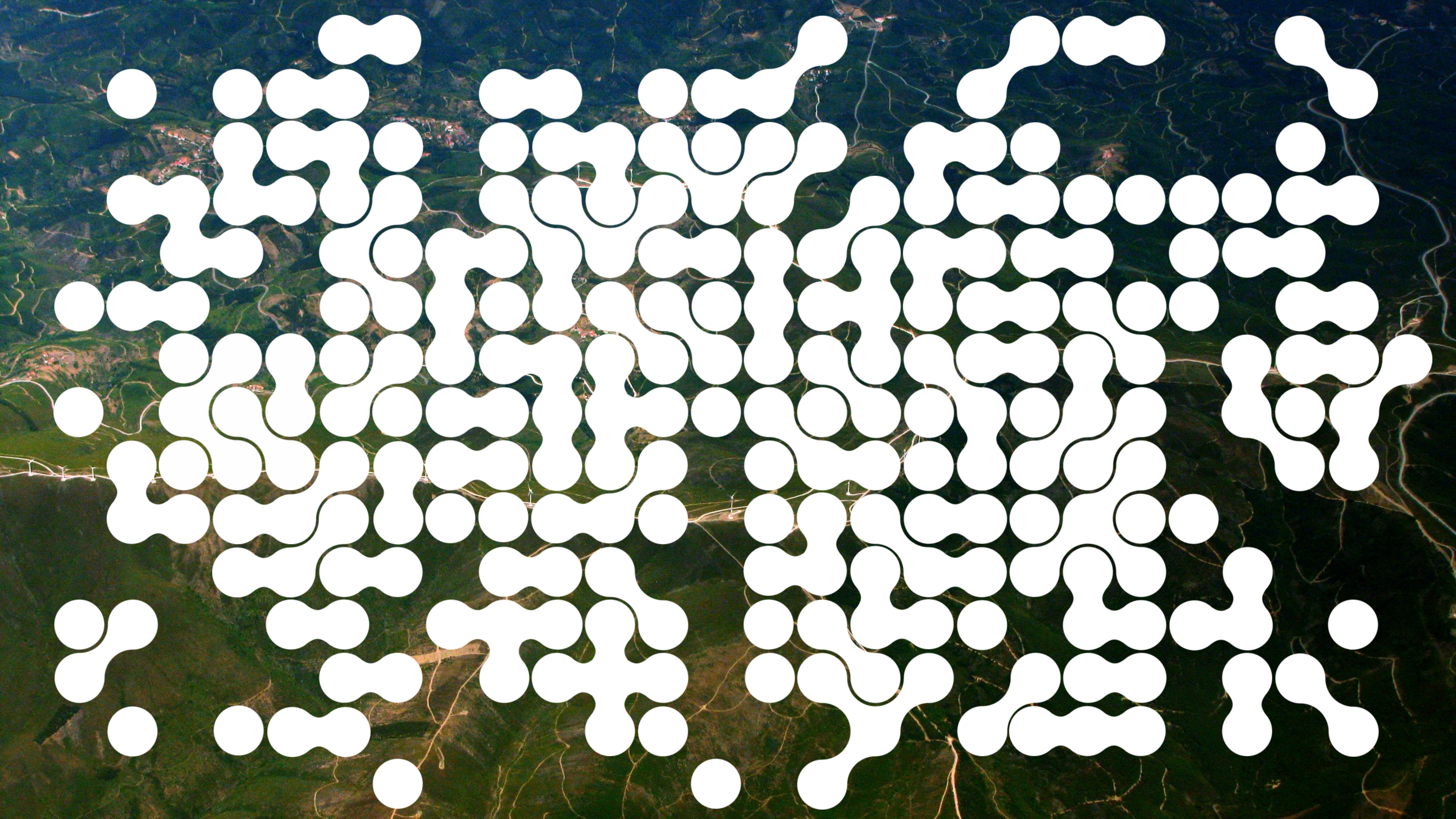


LandNetwork

INTERLIS Training

Fabian Mejia

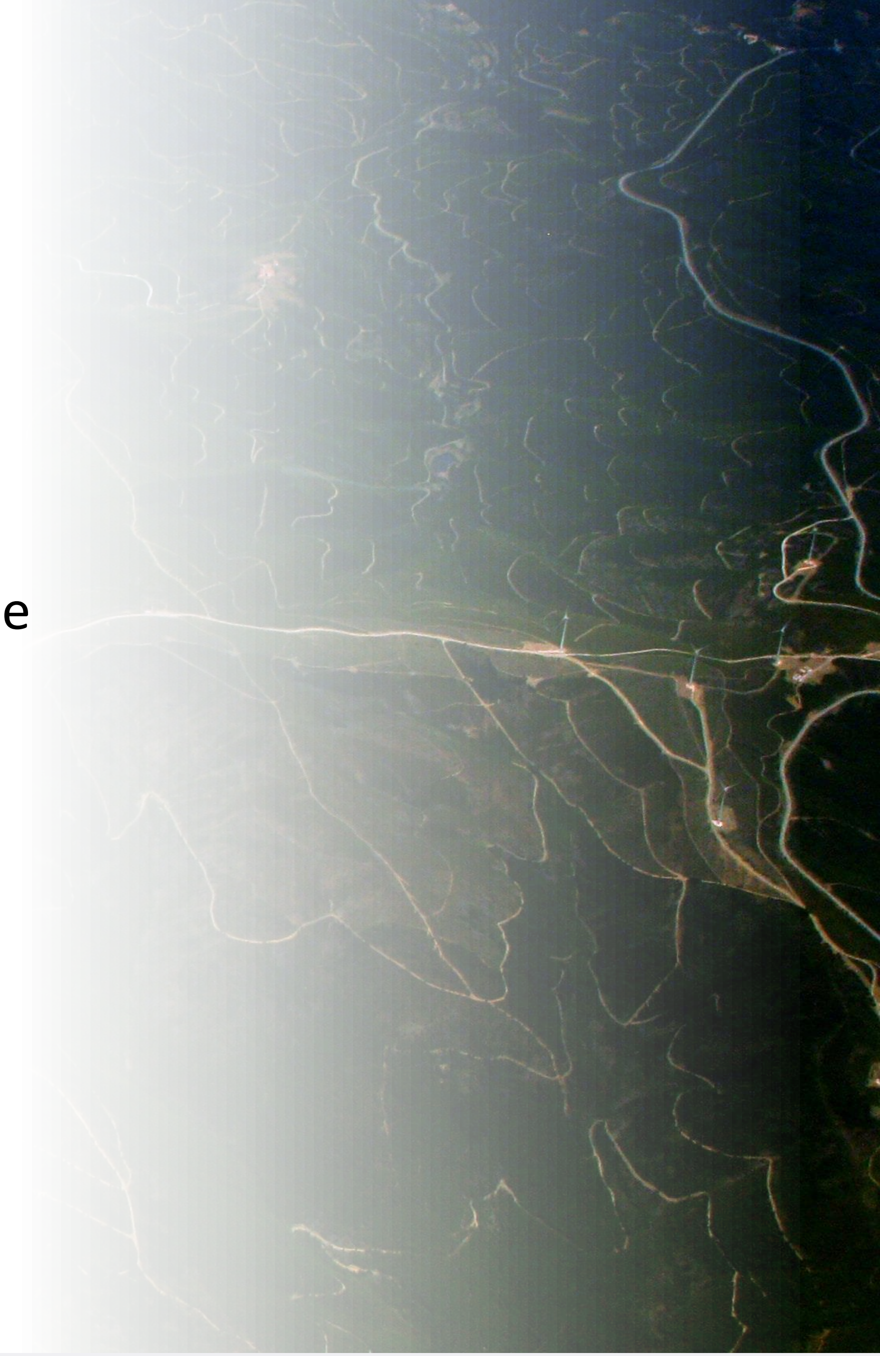
Land Administration Specialist

Fabian.mejia@landnetwork.ch

LandNetwork

Agenda

- Resources and tools
- Introduction to INTERLIS
- Language elements
    - Models
    - Model Repositories
    - Topic
    - Dependencies between models/topics
    - Classes
    - Structures
    - Generalization/specialization
    - Extension / inheritance
    - Attributes
    - Domains
    - Associations
    - Geometry types



Course Resources

Tools

- Text Editor: Notepad ++ (Recommended)
- QGis 3.16 +
- PostgreSQL 9.6 or upper + PostGIS 2.3 or upper
- Java VM (JRE 1.6 or upper)

Reference:

https://www.interlis.ch/download/interlis2/ili2-refman_2006-04-13_e.pdf

https://drive.infomaniak.com/app/share/189474/c1d19a50-43c8-4b34-b238-d4f7814f37c7

What is INTERLIS?

INTER Land **I**nformation **S**ystems

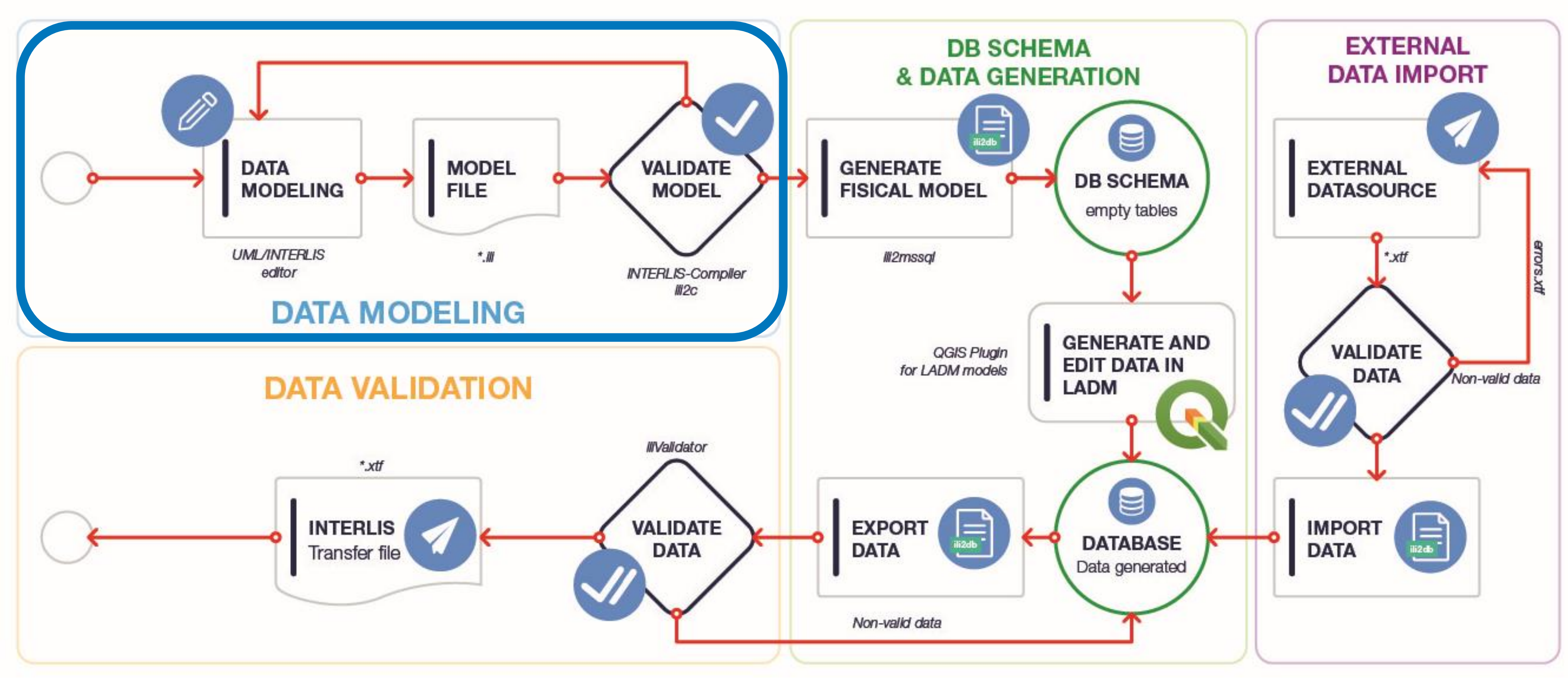
Conceptual schema language (.ili)

- A very precise standardized language at a conceptual level for the description of data models (schemas)
- System neutral (platform independent)
- Facilitates communication and understanding between IT and thematic specialists
- It is readable by both humans and machines
- Integrates different data types to be used on GIS (for example Geometries)

Data Transfer format (.itf/.xtf)

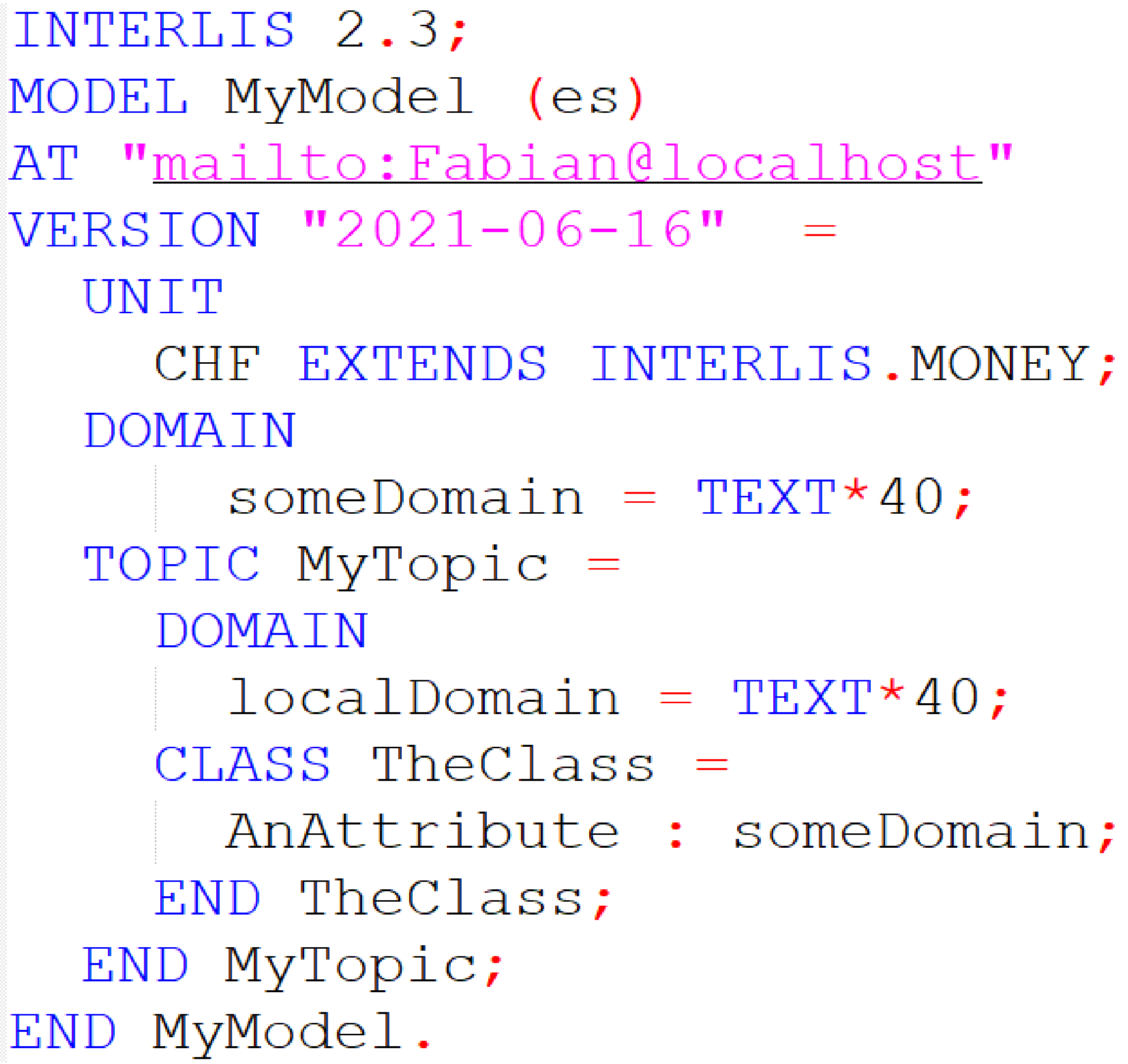
- The format (ITF or XTF) its derivated from the data model, through standardized rules
- Strict division between the transfer part and the modeling part (model driven approach)

Typical INTERLIS implementation Workflow

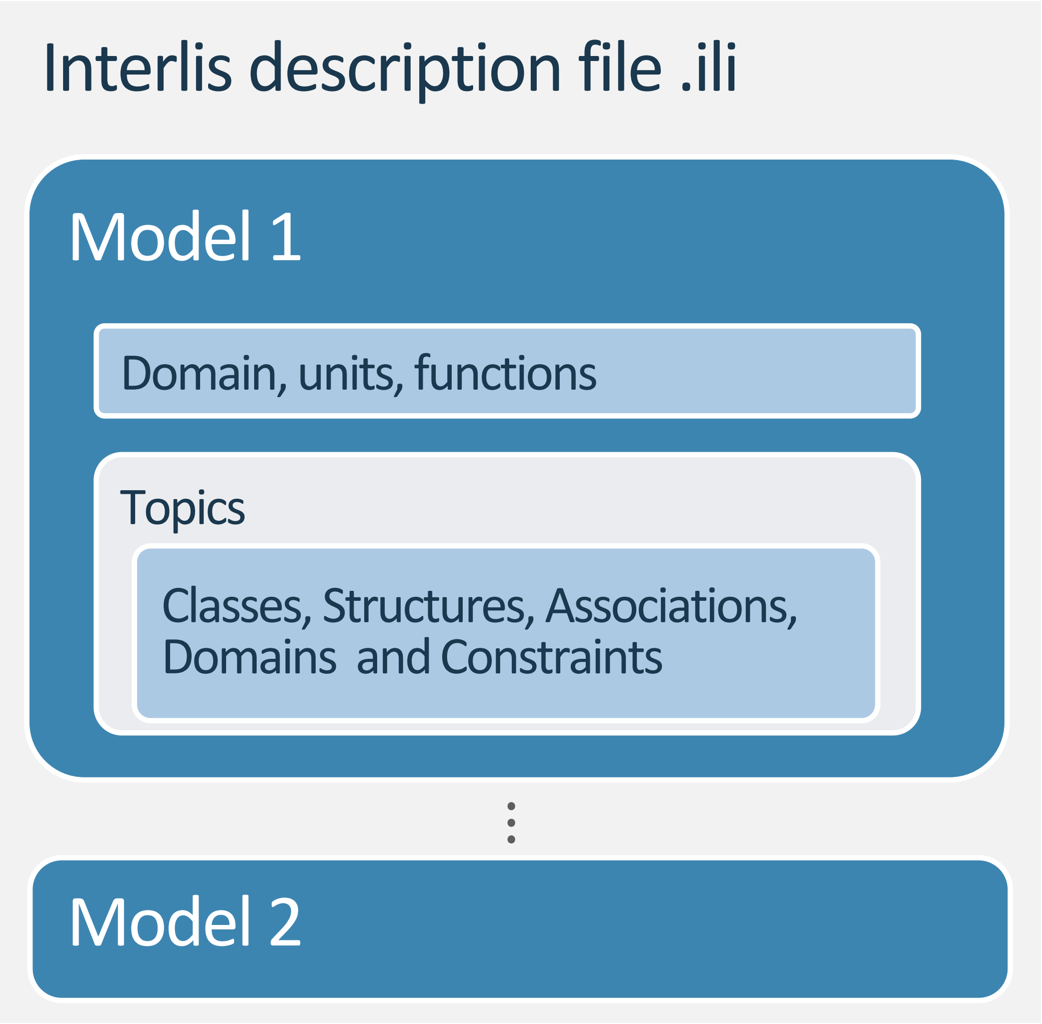


Mejía et al., 2017

General structure of an INTERLIS File



```
INTERLIS 2.3;
MODEL MyModel (es)
AT "mailto:Fabian@localhost"
VERSION "2021-06-16"  =
  UNIT
    CHF EXTENDS INTERLIS.MONEY;
  DOMAIN
      someDomain = TEXT*40;
  TOPIC MyTopic =
    DOMAIN
      localDomain = TEXT*40;
    CLASS TheClass =
      AnAttribute : someDomain;
    END TheClass;
  END MyTopic;
END MyModel.
```



Interlis description file .ili

Model 1

Domain, units, functions

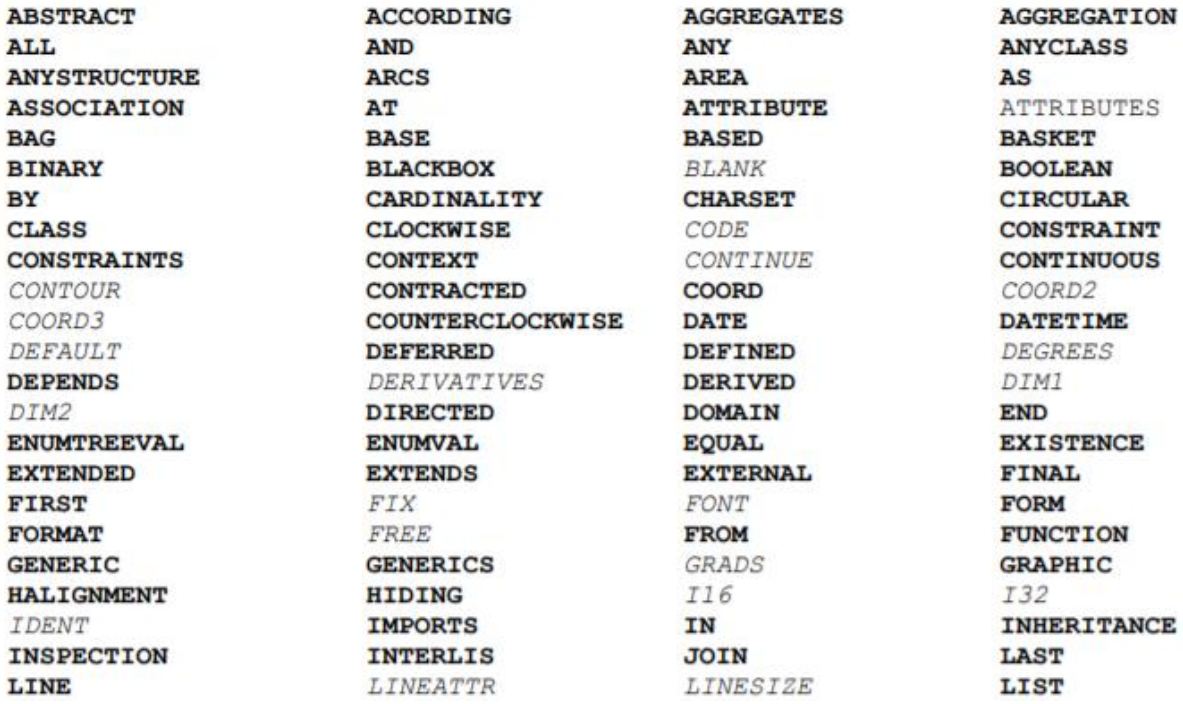
Topics

Classes, Structures, Associations, Domains and Constraints

⋮

Model 2

Reserved Keywords

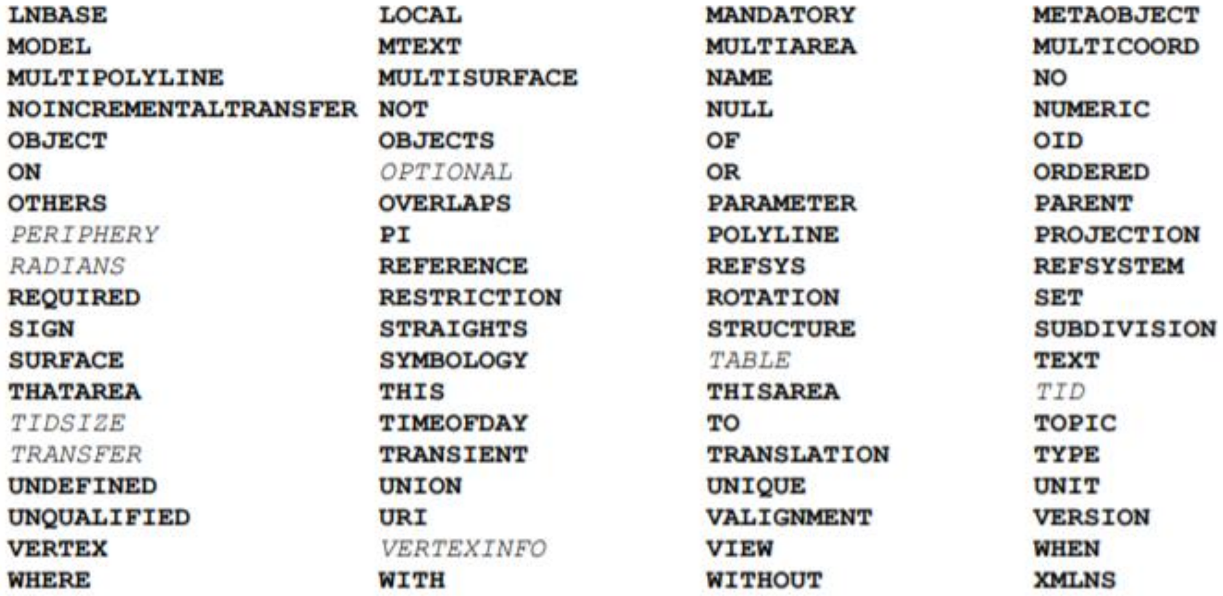


ABSTRACT
ALL
ANYSTRUCTURE
ASSOCIATION
BAG
BINARY
BY
CLASS
CONSTRAINTS
CONTOUR
COORD3
DEFAULT
DEPENDS
DIM2
ENUMTREEVAL
EXTENDED
FIRST
FORMAT
GENERIC
HALIGNMENT
IDENT
INSPECTION
LINE

ACCORDING
AND
ARCS
AT
BASE
BLACKBOX
CARDINALITY
CLOCKWISE
CONTEXT
CONTRACTED
COUNTERCLOCKWISE
DEFERRED
DERIVATIVES
DIRECTED
ENUMVAL
EXTENDS
FIX
FREE
GENERICS
HIDING
IMPORTS
INTERLIS
LINEATTR

AGGREGATES
ANY
AREA
ATTRIBUTE
BASED
BLANK
CHARSET
CODE
CONTINUE
COORD
DATE
DEFINED
DERIVED
DOMAIN
EQUAL
EXTERNAL
FONT
FROM
GRADS
I16
IN
JOIN
LINESIZE

AGGREGATION
ANYCLASS
AS
ATTRIBUTES
BASKET
BOOLEAN
CIRCULAR
CONSTRAINT
CONTINUOUS
COORD2
DATETIME
DEGREES
DIM1
END
EXISTENCE
FINAL
FORM
FUNCTION
GRAPHIC
I32
INHERITANCE
LAST
LIST



LNBASE
MODEL
MULTIPOLYLINE
NOINCREMENTALTRANSFER
OBJECT
ON
OTHERS
PERIPHERY
RADIANS
REQUIRED
SIGN
SURFACE
THATAREA
TIDSIZE
TRANSFER
UNDEFINED
UNQUALIFIED
VERTEX
WHERE

LOCAL
MTEXT
MULTISURFACE
NOT
OBJECTS
OPTIONAL
OVERLAPS
PI
REFERENCE
RESTRICTION
STRAIGHTS
SYMBOLOGY
THIS
TIMEOFDAY
TRANSIENT
UNION
URI
VERTEXINFO
WITH

MANDATORY
MULTIAREA
NAME
NULL
OF
OR
PARAMETER
POLYLINE
REFSYS
ROTATION
STRUCTURE
TABLE
THISAREA
TO
TRANSLATION
UNIQUE
VALIGNMENT
VIEW
WITHOUT

METAOBJECT
MULTICOORD
NO
NUMERIC
OID
ORDERED
PARENT
PROJECTION
REFSYSTEM
SET
SUBDIVISION
TEXT
TID
TOPIC
TYPE
UNIT
VERSION
WHEN
XMLNS

Model

- A model contains Units, Functions, Domains, Classes, Structures and Topic definitions
- Must have Issuer (AT) and Version attributes (Only for documentation purposes)

Syntaxis

```
ModelDef = 'MODEL' Model-Name
                  'AT' URI-String
                  'VERSION' Version-String
                  '='
                  { TopicDef }
              'END' Model-Name '.'
```

Example

```
INTERLIS 2.3;
MODEL UtilityNetwork
  AT "mailto:Fabian@localhost"
  VERSION "2021-06-22"  =
    DOMAIN
      Height = 0.00 .. 9000.00;
    TOPIC Water =
    END Water;
END UtilityNetwork.
```

Use definitions from another model

```
INTERLIS 2.3;
MODEL Earth
   AT "mailto:Fabian@localhost"
   VERSION "2021-06-16" =
      IMPORTS Units; !!see Appendix F of INTERLIS-Reference Manual
      DOMAIN
         Atmospheric_Pressure = 0.00 .. 99.00 [Units. atm];
      ...
END Earth.
```

- Use *IMPORTS* keyword followed by model name (no need to use path)
- Referenced object names are qualified (*Model.Topic.Class*)

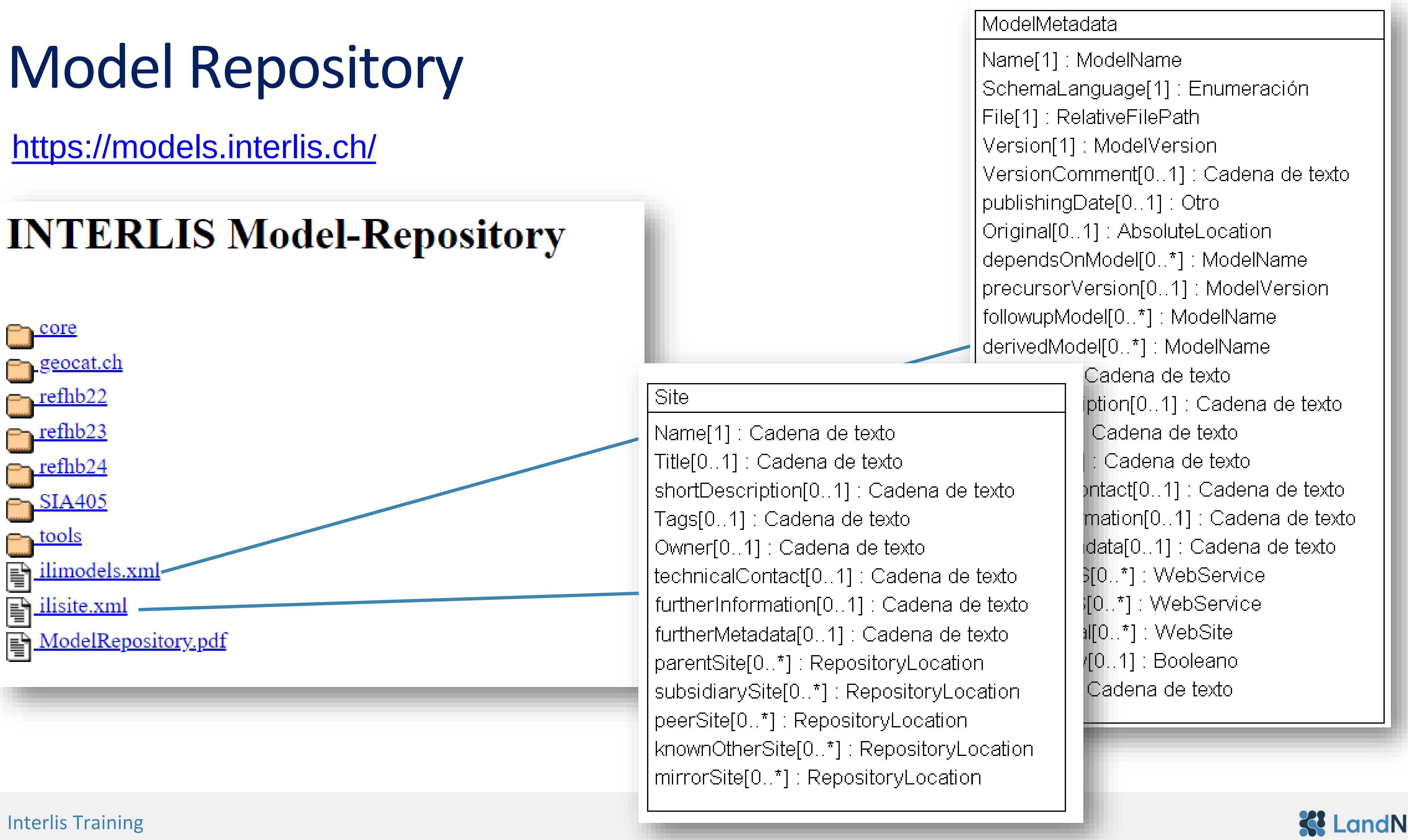
Use unqualified names

```
INTERLIS 2.3;
MODEL Earth
  AT "mailto:Fabian@localhost"
  VERSION "2021-06-16" =
    IMPORTS UNQUALIFIED Units;
    DOMAIN
      Atmospheric_Pressure = 0.00 .. 99.00 [atm]; !! No need to use Model Name
    ...
END Earth.
```

- Use *IMPORTS* ***UNQUALIFIED*** keyword enables the use of unqualified names

Model Repository

https://models.interlis.ch/



INTERLIS Model-Repository

- core
- geocat.ch
- refhb22
- refhb23
- refhb24
- SIA405
- tools
- ilimodels.xml
- ilisite.xml
- ModelRepository.pdf

ModelMetadata

Name[1] : ModelName
SchemaLanguage[1] : Enumeración
File[1] : RelativeFilePath
Version[1] : ModelVersion
VersionComment[0..1] : Cadena de texto
publishingDate[0..1] : Otro
Original[0..1] : AbsoluteLocation
dependsOnModel[0..*] : ModelName
precursorVersion[0..1] : ModelVersion
followupModel[0..*] : ModelName
derivedModel[0..*] : ModelName
Cadena de texto
iption[0..1] : Cadena de texto
Cadena de texto
| : Cadena de texto
ontact[0..1] : Cadena de texto
mation[0..1] : Cadena de texto
data[0..1] : Cadena de texto
S[0..*] : WebService
S[0..*] : WebService
al[0..*] : WebSite
y[0..1] : Booleano
Cadena de texto

Site

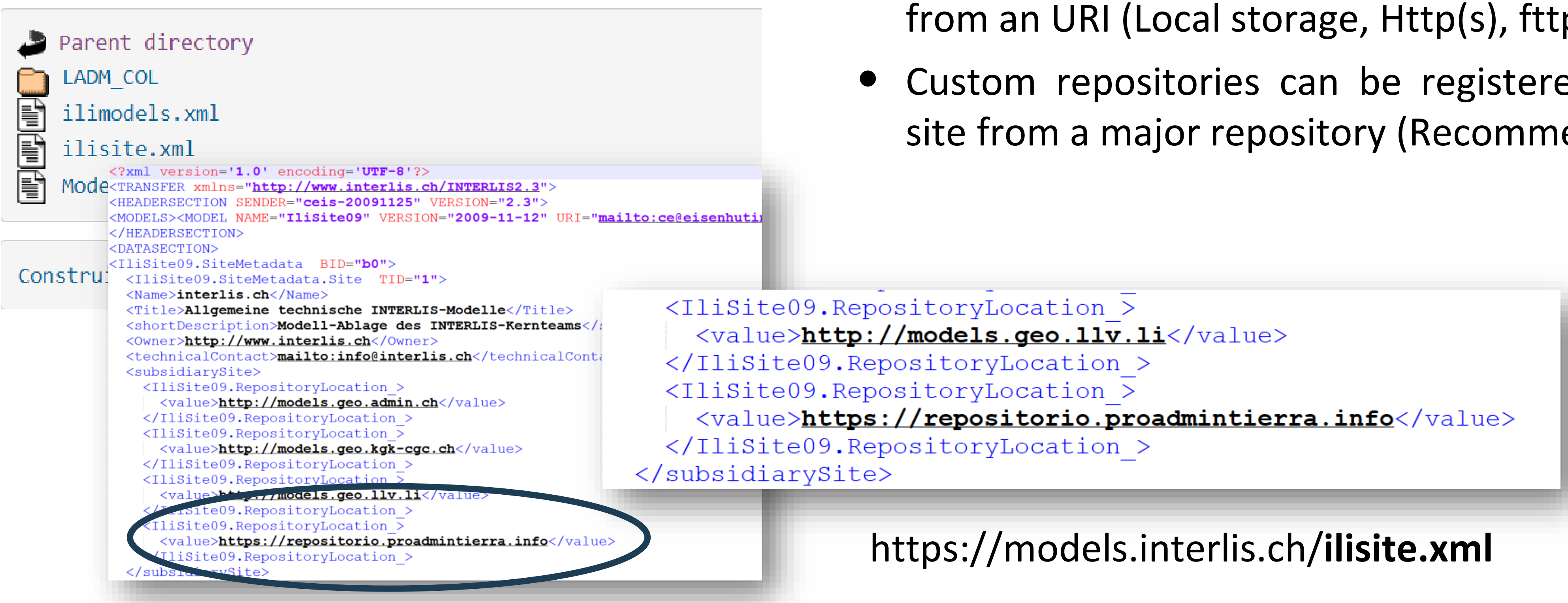
Name[1] : Cadena de texto
Title[0..1] : Cadena de texto
shortDescription[0..1] : Cadena de texto
Tags[0..1] : Cadena de texto
Owner[0..1] : Cadena de texto
technicalContact[0..1] : Cadena de texto
furtherInformation[0..1] : Cadena de texto
furtherMetadata[0..1] : Cadena de texto
parentSite[0..*] : RepositoryLocation
subsidiarySite[0..*] : RepositoryLocation
peerSite[0..*] : RepositoryLocation
knownOtherSite[0..*] : RepositoryLocation
mirrorSite[0..*] : RepositoryLocation

Building a local model repository

Colombian Example https://repositorio.proadmintierra.info/

Model Repository LADM_COL

https://repositorio.proadmintierra.info /



```
Parent directory
LADM_COL
ilimodels.xml
ilisite.xml
Mode
Constru
```

```
<?xml version='1.0' encoding='UTF-8'?>
<TRANSFER xmlns="http://www.interlis.ch/INTERLIS2.3">
<HEADERSECTION SENDER="ceis-20091125" VERSION="2.3">
<MODELS><MODEL NAME="IliSite09" VERSION="2009-11-12" URI="mailto:ce@eisenhuti
</HEADERSECTION>
<DATASECTION>
<IliSite09.SiteMetadata  BID="b0">
  <IliSite09.SiteMetadata.Site  TID="1">
  <Name>interlis.ch</Name>
  <Title>Allgemeine technische INTERLIS-Modelle</Title>
  <shortDescription>Modell-Ablage des INTERLIS-Kernteams</
  <Owner>http://www.interlis.ch</Owner>
  <technicalContact>mailto:info@interlis.ch</technicalConta
  <subsidiarySite>
    <IliSite09.RepositoryLocation_>
      <value>http://models.geo.admin.ch</value>
    </IliSite09.RepositoryLocation_>
    <IliSite09.RepositoryLocation_>
      <value>http://models.geo.kgk-cgc.ch</value>
    </IliSite09.RepositoryLocation_>
    <IliSite09.RepositoryLocation_>
      <value>http://models.geo.llv.li</value>
    </IliSite09.RepositoryLocation_>
    <IliSite09.RepositoryLocation_>
      <value>https://repositorio.proadmintierra.info</value>
    </IliSite09.RepositoryLocation_>
  </subsidiarySite>
```

```
  <IliSite09.RepositoryLocation_>
    <value>http://models.geo.llv.li</value>
  </IliSite09.RepositoryLocation_>
  <IliSite09.RepositoryLocation_>
    <value>https://repositorio.proadmintierra.info</value>
  </IliSite09.RepositoryLocation_>
</subsidiarySite>
```

- A model repository can be at any place accessible from an URI (Local storage, Http(s), fttp(s))
- Custom repositories can be registered as subsidiary site from a major repository (Recommended)

https://models.interlis.ch/**ilisite.xml**

Topic

- Contains all definitions to describe a part of the modeled reality.
- A topic can contain *Units, Functions, Domains, Classes, Structures, Associations, Constraints* and *Views* definitions

Syntaxis

```
TopicDef =  'TOPIC' Topic-Name ‘=‘
                 Definitions
               'END' Topic-Name ';'
```

Example

```
INTERLIS 2.3;
MODEL UtilityNetwork
  AT "mailto:Fabian@localhost"
  VERSION "2021-06-22"  =
    DOMAIN
      Height = 0.00 .. 9000.00;
    TOPIC WaterNetwork =
      CLASS Pipeline =
      END Pipeline;
    END WaterNetwork;
END UtilityNetwork.
```

Extending Topics

- A global model defines common rules for al parties, in some cases local rules need to be applied extending Topics.

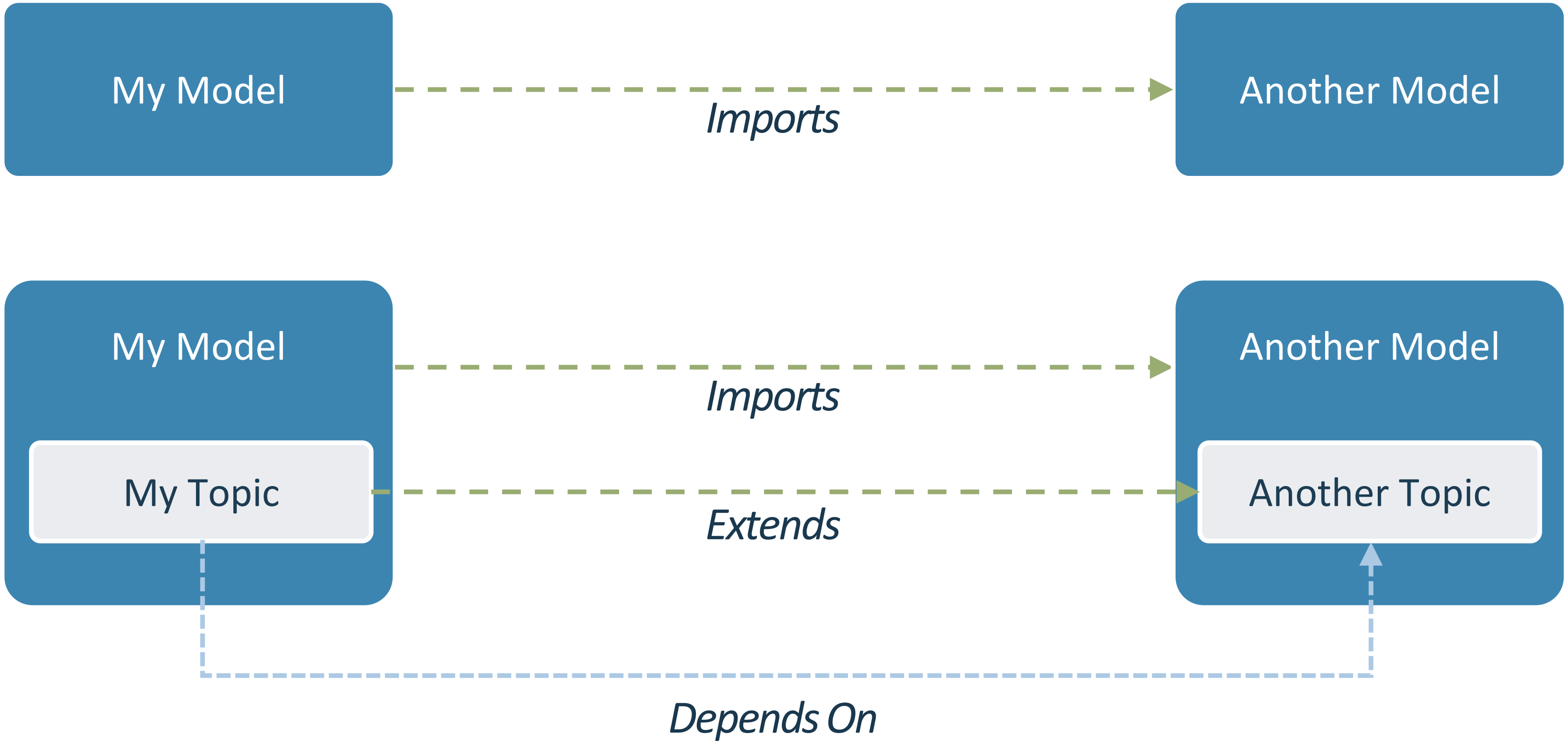
Syntaxis

```
TopicDef =   'TOPIC' Topic-Name
                'EXTENDS' TopicRef '='
                    Definitions
             'END' Topic-Name ';'.
```

Example

```
TOPIC LocalWater EXTENDS WaterNetwork =
    CLASS Pipeline (EXTENDED) =
        !! + includes attributes of class Pipeline in WaterNetwork
        LastMaintenance: INTERLIS.XMLDate;
    END Pipeline;
END LocalWater;
```

Dependencies between Models and Topics



Classes

- Classes defines a general structure of attributes, constraints, and parameters applicable to all objects of the same type.

Syntaxis

```
ClassDef =   'CLASS' Class-Name '='
               { AttributeDef }
             'END' Class-Name ';'.
```

Example

```
CLASS Pipeline =
   Type : TEXT*20;
   Size : 0 .. 50 [in];
END Pipeline;
```

Types of Classes

Classes can be

- Concrete
- Abstract
- Final
- Derivate/Extended (*Discussion*)

Structures

- Structures as classes does, can be concretes, abstracts, finals, or extended.

Syntaxis

```
StructureDef = 'STRUCTURE' Struct-Name '='
                  { AttributeDef }
               'END' Struct-Name ';'.
```

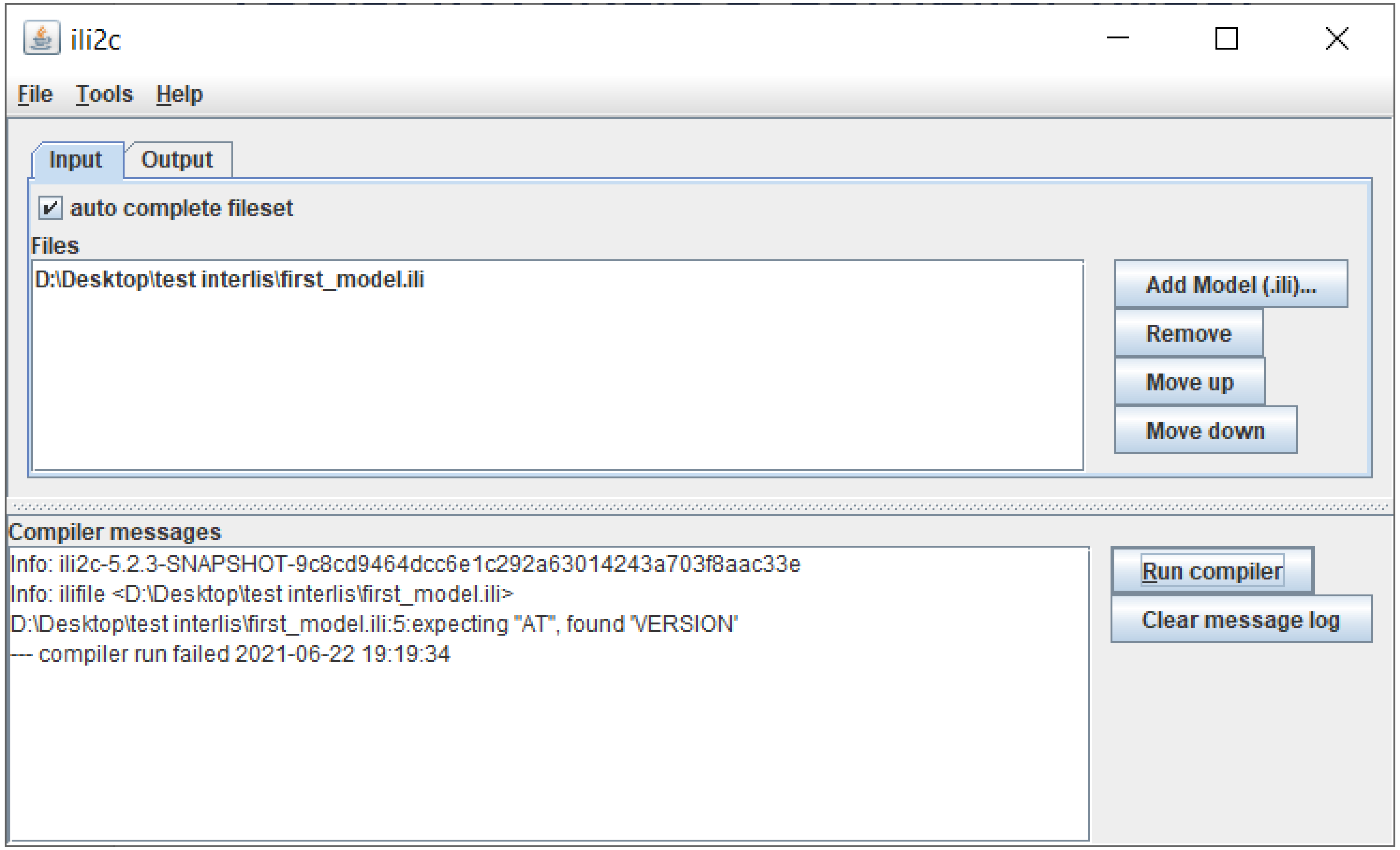
Example

```
STRUCTURE ParcelAddress =
   StreetName : TEXT*40;
   Number : TEXT*12;
END ParcelAddress;
CLASS Parcel =
   Number : ParcelAddress;
END Parcel;
CLASS CantonParcels =
   HeritageParcels : BAG {0..*} OF ParcelAddress;
END CantonParcels;
```

Creating and checking models

Tools: INTERLIS 2 Compiler (ili2c)

https://github.com/claeis/ili2c



☑ auto complete fileset

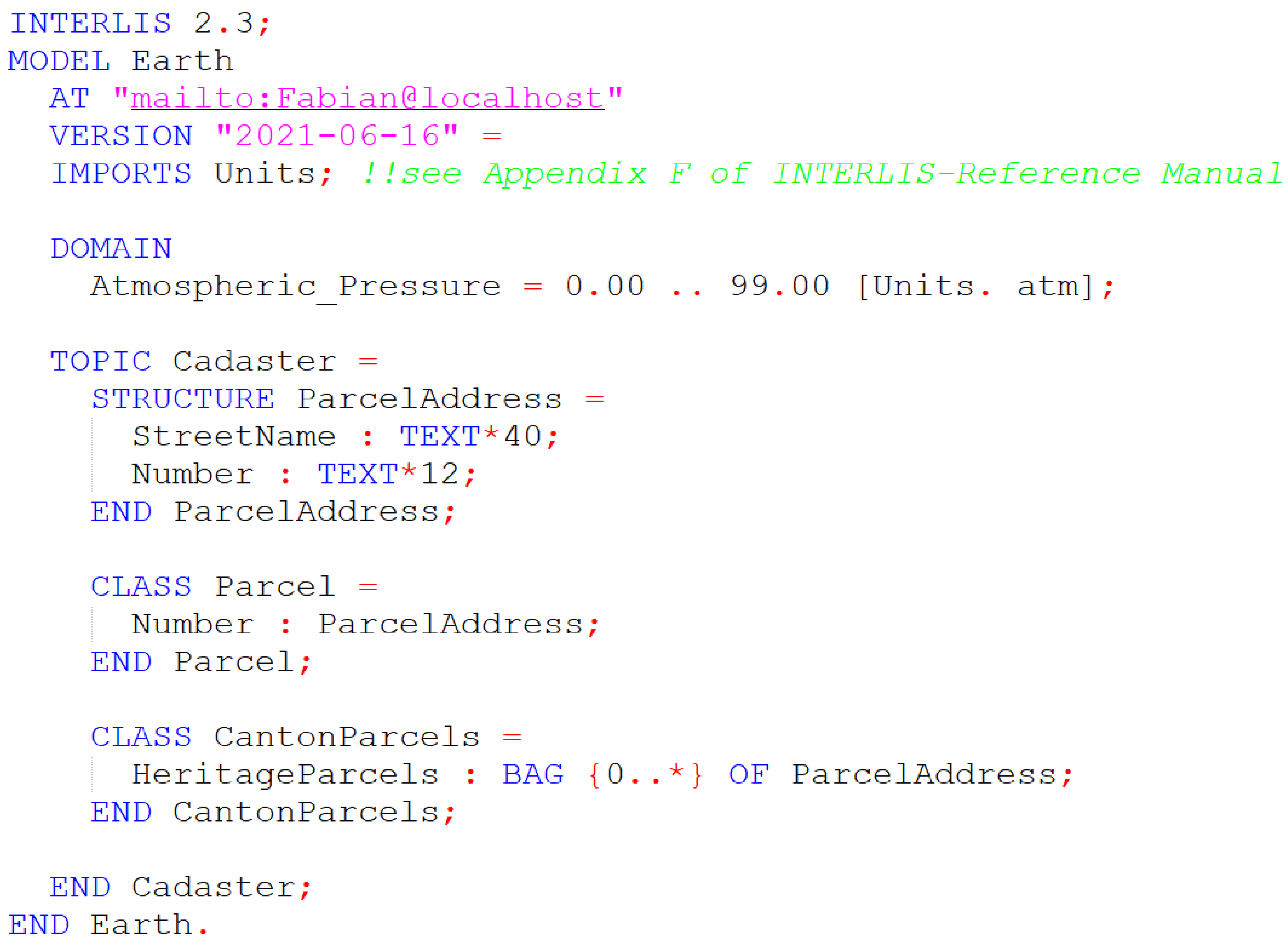
Files

D:\Desktop\test interlis\first_model.ili

Compiler messages

```
Info: ili2c-5.2.3-SNAPSHOT-9c8cd9464dcc6e1c292a63014243a703f8aac33e
Info: ilifile <D:\Desktop\test interlis\first_model.ili>
D:\Desktop\test interlis\first_model.ili:5:expecting "AT", found 'VERSION'
--- compiler run failed 2021-06-22 19:19:34
```

Create and check a model



```
INTERLIS 2.3;
MODEL Earth
  AT "mailto:Fabian@localhost"
  VERSION "2021-06-16" =
  IMPORTS Units; !!see Appendix F of INTERLIS-Reference Manual

  DOMAIN
    Atmospheric_Pressure = 0.00 .. 99.00 [Units. atm];

  TOPIC Cadaster =
    STRUCTURE ParcelAddress =
      StreetName : TEXT*40;
      Number : TEXT*12;
    END ParcelAddress;

    CLASS Parcel =
      Number : ParcelAddress;
    END Parcel;

    CLASS CantonParcels =
      HeritageParcels : BAG {0..*} OF ParcelAddress;
    END CantonParcels;

  END Cadaster;
END Earth.
```

Generalization and specialization of a class

Syntaxis

ClassDef = 'CLASS' Class-Name
'**EXTENDS**' ClassRef '='
'END' Class-Name ';'.

Example

CLASS Creature =
END Creature;

CLASS Animal **EXTENDS** Creature =
END Animal;

Extending Classes

Syntaxis

```
ClassDef =    'CLASS' Class-Name
                 '(' EXTENDED ')' '='
              'END' Class-Name ';'.
```

Example

```
TOPIC WaterNetwork =
    CLASS Pipeline =
    END Pipeline;
 END WaterNetwork;

 TOPIC LocalWater EXTENDS WaterNetwork =
    CLASS Pipeline (EXTENDED) =
        LastMaintenance: INTERLIS.XMLDate;
    END Pipeline;
 END LocalWater;
```

Enforce specialization

Syntaxis

```
ClassDef =    'CLASS' Class-Name
                 '(ABSTRACT)'
                 [ 'EXTENDS' ClassRef ] '='
              'END' Class-Name ';'.
```

Example

```
CLASS Creature (ABSTRACT) =
END Creature;

CLASS Animal EXTENDS Creature =
END Animal;
```

Prevent specialization

Syntaxis

```
ClassDef =    'CLASS' Class-Name
                 '(FINAL)'
                 [ 'EXTENDS' ClassRef ] '='
              'END' Class-Name ';'.
```

Example

```
CLASS Creature (FINAL) =
END Creature;

CLASS Animal EXTENDS Creature =
END Animal;
```

!! Error: Creature cannot be extended

Attributes

Syntaxis

```
AttributDef =
  Attribute-Name : [MANDATORY]
    Type
    | DomainRef;

DomainRef =
      [ Model-Name '.' [ Topic-Name '.' ] ] Domain-Name
```

Example

```
CLASS Person =
   Name : TEXT*40;
   ID : MANDATORY TEXT*10;
   Genre : ( Male, Female );
   HasChildren: BOOLEAN;
END Person;
```

Domain of values/Data types

- Character (TextType)
- Number (NumericType)
- Date and Time (FormattedType)
- Enumeration (EnumerationType, BooleanType)
- Identifier (OIDType)
- Blackbox (BlackboxType)
- Coordinate (CoordinateType)
- Line (PolylineType)
- Polygon (PolylineType)

Text Data types

- Name : TEXT*40;
- Title : TEXT;
- Description : MTEXT;
- InterlisModelName : NAME;
- EmailAddress : URI;

Over-Restricting attributes

```
CLASS Site =
   Name : TEXT*120;
END Site;

CLASS Museum EXTENDS Site =
   Name (EXTENDED) : TEXT*80;
END Museum;
```

Enumerations

```
EnumerationType = Enumeration .
Enumeration = '(' EnumElement { ',' EnumElement } [ ':' 'FINAL' ]')'.
EnumElement = EnumElement-Name [Sub-Enumeration].
BooleanType = 'BOOLEAN'.
```

Example

```
Color : (blue
        , green
        , red (orange, carmine, darkred));
Validated : BOOLEAN;
```

Specialization of an enumeration

```
DOMAIN
Color = (blue
        , green
        , red);
ColorExact EXTENDS Color = (red (orange, carmine, darkred));
```

Extending of an enumeration

```
DOMAIN
Color = (blue
        , green
        , red);
MoreColors EXTENDS Color = (Black, White);
```

Associations

- association (--): Relationship between independent objects
- aggregation (-<>): Relationship between Parts and a Whole. A Part can be part of multiple Wholes.
- composition (-<#>): Relationship between Parts and a Whole. A Part can only be part of a single Whole.

```
AssociationDef = 'ASSOCIATION' '='
                       { RoleDef }
                    'END' ';'.
RoleDef = Role-Name '--' ClassRef ';'.
```

Example

```
CLASS School ... ;
CLASS Person ... ;

ASSOCIATION =
   primarySchool -- School;
   director -- Person;
END;
```

Cardinality and Force

Cardinality (number of objects)

```
CLASS School =
END School;
CLASS Person =
END Person;


ASSOCIATION =
   primarySchool -- {0..*} School;
   director -- {1} Person;
END;
```

Composition

```
ASSOCIATION =
   car -<#> {1} Vehicle;
   wheel -- {4} Wheel;
END;
```

Ordering related objects

```
ASSOCIATION =
   line -- {1}  Polyline;
   vertex (ORDERED) -- {2..*} Point;
END;
```

Associations with attributes

```
ASSOCIATION =
   employer -- Company;
   employee -- Person;
   Salary : 1500 .. 15000 [CHF];
END;
```

Coordinates

- INTERLIS doesn’t know about EPSG
- The CRS/EPSG code is not in the transfer (because it is constant for a given model/geometry attribute)
- The CRS is defined/used in the model
- *A COORD domain references a CRS*

Example projected coordinates

```
DOMAIN
    CoordLV03 = COORD
        460000.000 .. 870000.000 [m],
        45000.000 .. 310000.000 [m],
        ROTATION 2 -> 1;
```

Geographic coordinates

```
REFSYSTEM BASKET
    BCoordSys ~ CoordSys.CoordsysTopic
    OBJECTS OF GeoEllipsoidal : WGS84;

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

Thank you

Fabian.mejia@landnetwork.ch