The General Feature Model (GFM)

GML Application schema

Contents

- In this module you will learn about:
  - The „General Feature Model“ (ISO 19109)
  - Design and structure of application schemas conformant to the GFM
  - Hints for defining UML Application schemas in UML tools (and automatically generate according GML Application schemas)

The „General Feature Model“ (important)

- The General feature model (GFM) is a meta-model for developing conceptual models of feature types and their properties. It defines the concept of feature type, feature attribute, feature association and feature operation. It also serves as a meta-model for feature catalogues by providing the structure for representing the semantics of geographic information in these terms.

- As there is no GFM-language and the conceptual schema must be expressed in a CSL, the concepts of the GFM must be mapped into the concepts of the used conceptual schema language. The structure and concepts of the GFM must be kept in mind when making the conceptual model. The result can be documented in terms of GFM in a feature catalogue.

- The application schema contains instances of types defined in the General feature model. It is used by experts in business requirements for the geographic information domain to develop application schemas for specific applications. For example, the General feature model defines the concept of feature type while an application schema defines specific feature types such as Road or Lake.

(see ISO DIS 19101:2001, section 8.6.7)

Purpose of General Feature Model (important) (GFM)

The GFM is a model of the concepts required to classify a view of the real world. It is expressed in a CSL (conceptual schema language), that is in UML class diagrams, but it could be in any CSL. UML has its own model of concepts (metamodel).

As both the GFM and the UML metamodel deal with classification, the concepts are very similar. There is one big difference. The concepts in the GFM establish a basis for the classification of features whereas the UML-metamodel provides a basis for classification of any kind.

(see ISO19109)

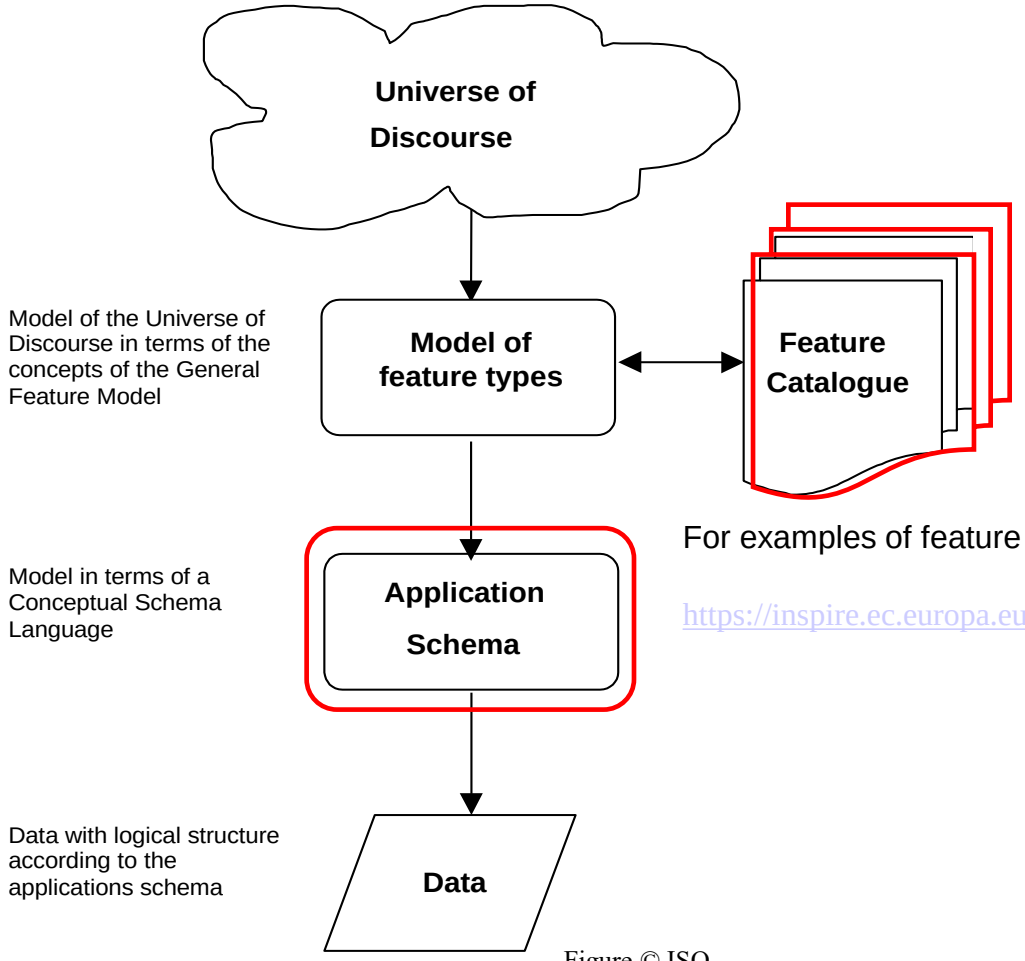
ISO 19109 Rules for Application Schemas Scope (important)

GFM as metamodel provides the framework for

- √ conceptual modelling of features and their properties from a universe of discourse;
- √ definition of application schemas;
- √ use of the conceptual schema language for application schemas;
- √ transition from the concepts in the conceptual model to the data types in the application schema;
- √ integration of standardized schemas from other ISO geographic information standards with the application schema.

Application Schema Development (important) (ISO 19109)

- Start by clearly defining the scope
- Model in terms of the 'General Feature Model'
- Derive an 'Application Schema'
- Encode data



For examples of feature catalogues see:

https://inspire.ec.europa.eu/data-model/approved/r4618/fc/

Figure © ISO

Feature Catalogs (important)

- Abstraction of reality represented in one or more sets of geographic data as a defined classification of phenomena,
  The basic level of classification is the feature type.

- Catalog containing definitions and descriptions of the feature types, feature attributes, and feature associations occurring in one or more sets of geographic data, together with any feature operations that may be applied.

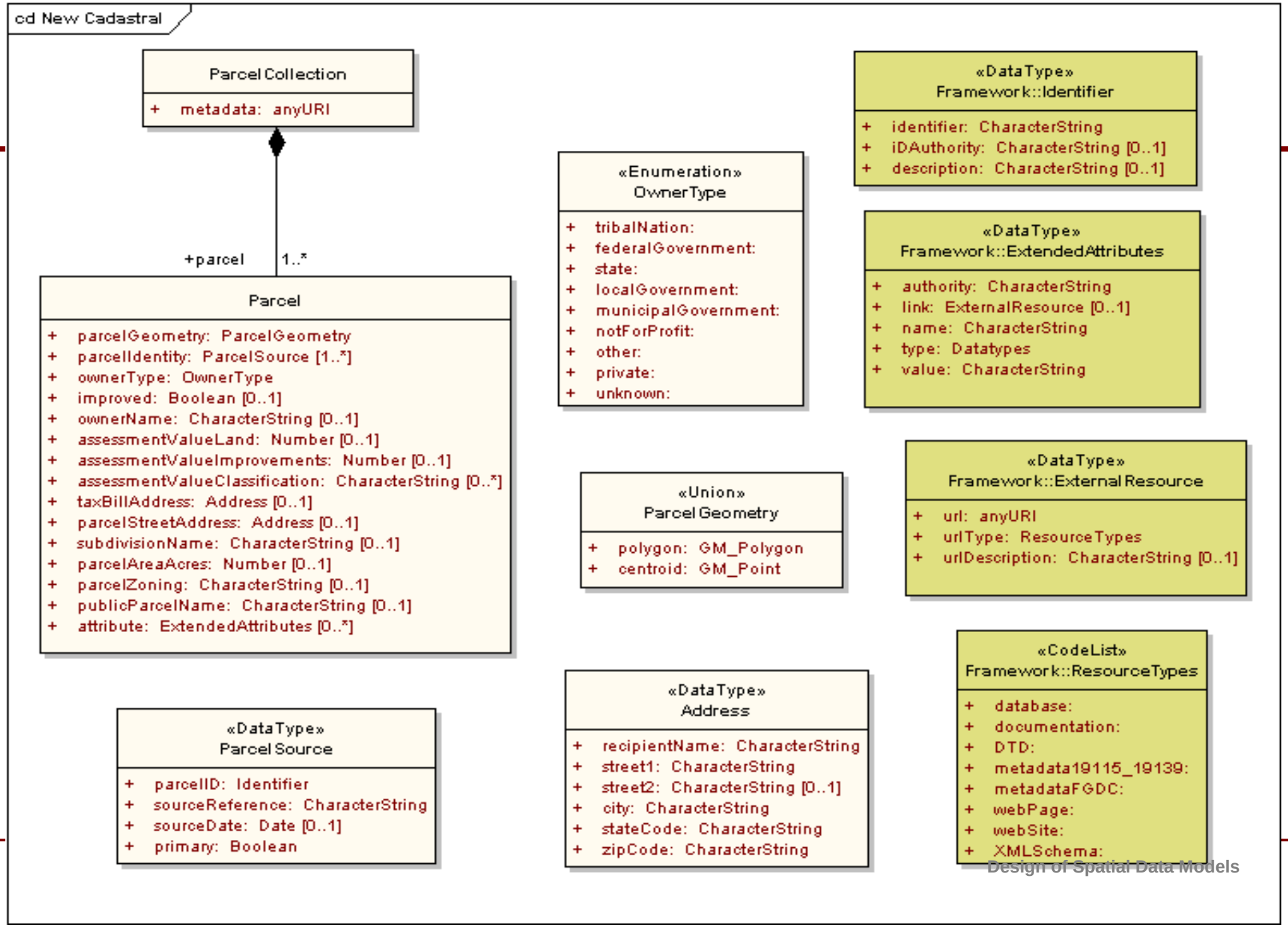
- ISO 19110 methodology for "cataloging feature" provides the basis for the description of feature types that can be applied across domains (uniform meaning across different user communities is arranged).

	Feature Type	Class of real world phenomena with common properties	
11	Name	Text string that uniquely identifies the feature type within the catalogue	
12	Definition	Definition of the feature type in a natural language	
13	Code	Code that uniquely identifies the feature type within a catalogue	
14	Aliases	Name(s) of equivalent feature term(s)	
15	Feature Operation Names	Operations that every instance of this feature type may perform	
16	Feature Attribute Names	Characteristic(s) of the feature type	
17	Feature Association Names	Association(s) between instances of this feature type and instances of the same or a different feature type	
18	Subtype of	Identifies one or more feature types from which the subject feature type inherits all properties, including feature operations, feature attributes, and feature associations	

Figure © ISO

Application Schemas (important)

- An application schema
  - Provides the formal description of the data structure and content required by one or more applications
  - Specifies the operations for manipulating and processing data by these applications
  - Is expressed using a conceptual schema language (e.g., Unified Modeling Language UML)


cd New Cadastral
ParcelCollection
+ metadata: anyURI
+parcel
1..*
Parcel
+ parcelGeometry: ParcelGeometry
+ parcelIdentity: ParcelSource [1..*]
+ ownerType: OwnerType
+ improved: Boolean [0..1]
+ ownerName: CharacterString [0..1]
+ assessmentValueLand: Number [0..1]
+ assessmentValueImprovements: Number [0..1]
+ assessmentValueClassification: CharacterString [0..*]
+ taxBillAddress: Address [0..1]
+ parcelStreetAddress: Address [0..1]
+ subdivisionName: CharacterString [0..1]
+ parcelAreaAcres: Number [0..1]
+ parcelZoning: CharacterString [0..1]
+ publicParcelName: CharacterString [0..1]
+ attribute: ExtendedAttributes [0..*]
«Enumeration»
OwnerType
+ tribalNation:
+ federalGovernment:
+ state:
+ localGovernment:
+ municipalGovernment:
+ notForProfit:
+ other:
+ private:
+ unknown:
«DataType»
Framework::Identifier
+ identifier: CharacterString
+ iDAuthority: CharacterString [0..1]
+ description: CharacterString [0..1]
«DataType»
Framework::ExtendedAttributes
+ authority: CharacterString
+ link: ExternalResource [0..1]
+ name: CharacterString
+ type: Datatypes
+ value: CharacterString
«Union»
ParcelGeometry
+ polygon: GM_Polygon
+ centroid: GM_Point
«DataType»
Framework::ExternalResource
+ url: anyURI
+ urlType: ResourceTypes
+ urlDescription: CharacterString [0..1]
«DataType»
ParcelSource
+ parcelID: Identifier
+ sourceReference: CharacterString
+ sourceDate: Date [0..1]
+ primary: Boolean
«DataType»
Address
+ recipientName: CharacterString
+ street1: CharacterString
+ street2: CharacterString [0..1]
+ city: CharacterString
+ stateCode: CharacterString
+ zipCode: CharacterString
«CodeList»
Framework::ResourceTypes
+ database:
+ documentation:
+ DTD:
+ metadata19115_19139:
+ metadataFGDC:
+ webPage:
+ webSite:
+ XMLSchema:

GFM – a 'metamodel' for feature types (important)

- A feature type is defined by:
  - [1] a name and description
  - [2] its attributes/properties
  - [3] association roles
  - [4] defined behaviour of the feature type.
- Additional concepts:
  - [5] feature associations between the feature type and itself or other feature types
  - [6] generalization and specialization relationships to other feature types
  - [7] constraints on the feature type.

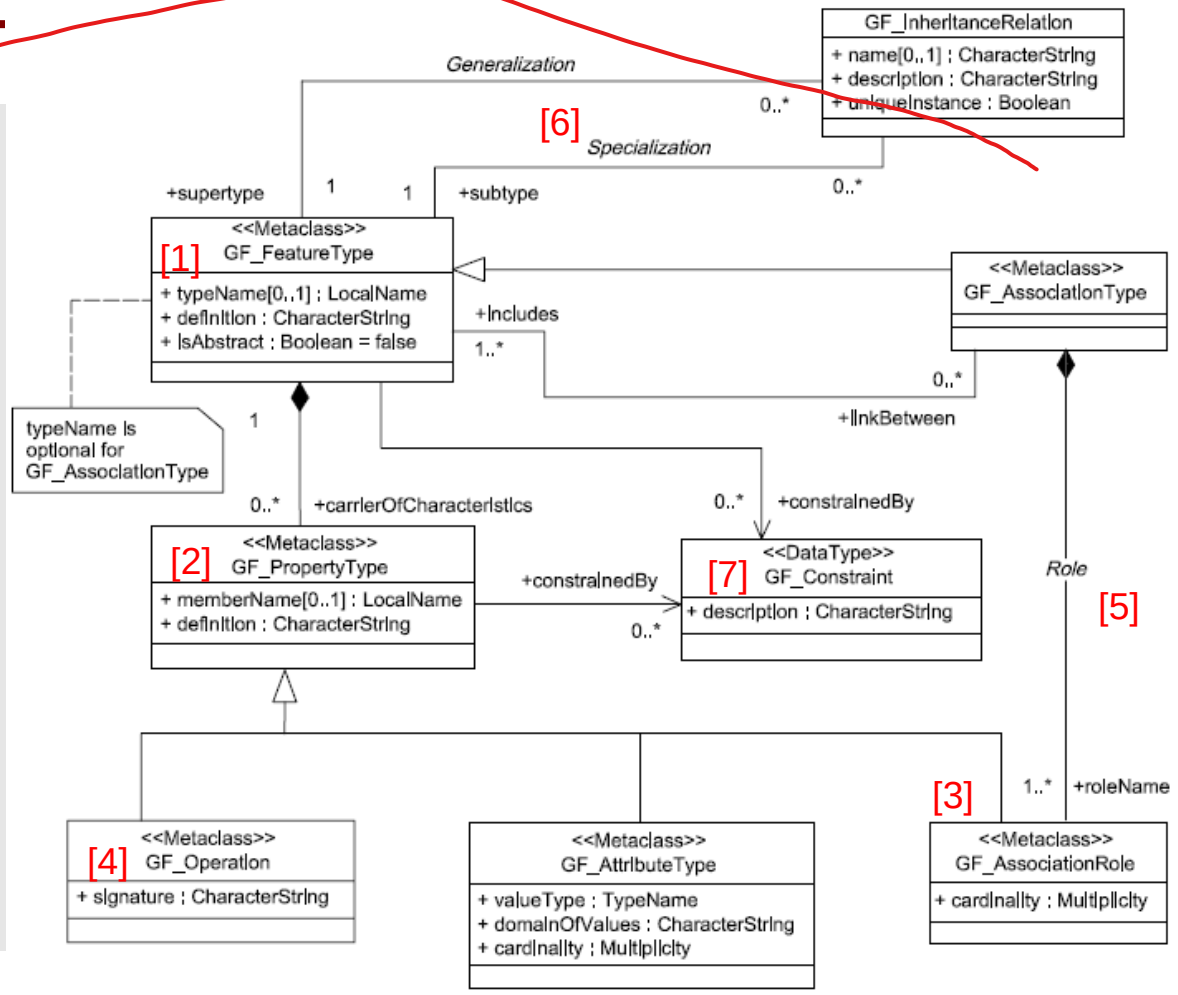
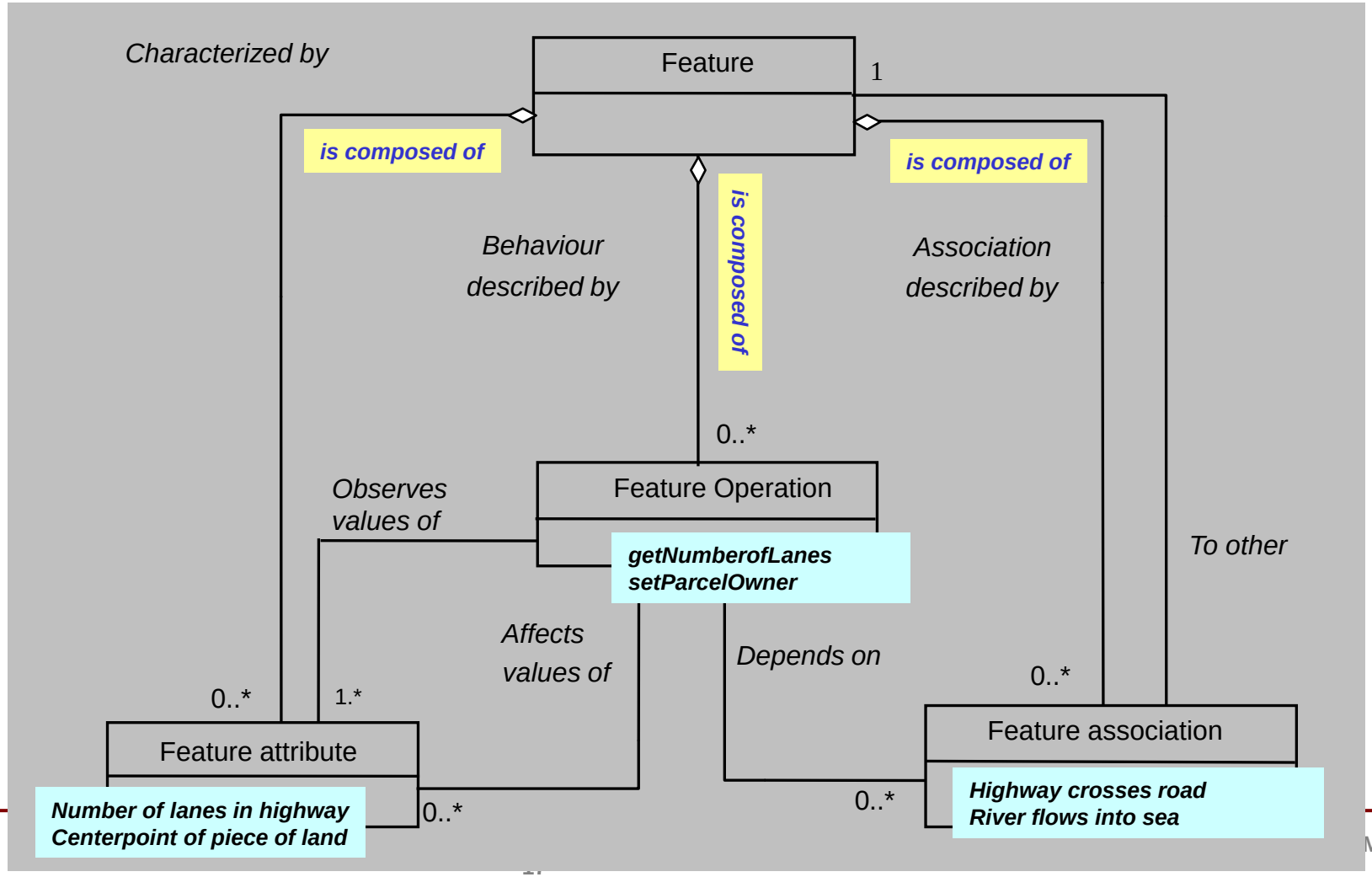


Figure 5 — Extract from the General Feature Model

Data Modelling in adherence to the GFM (important)

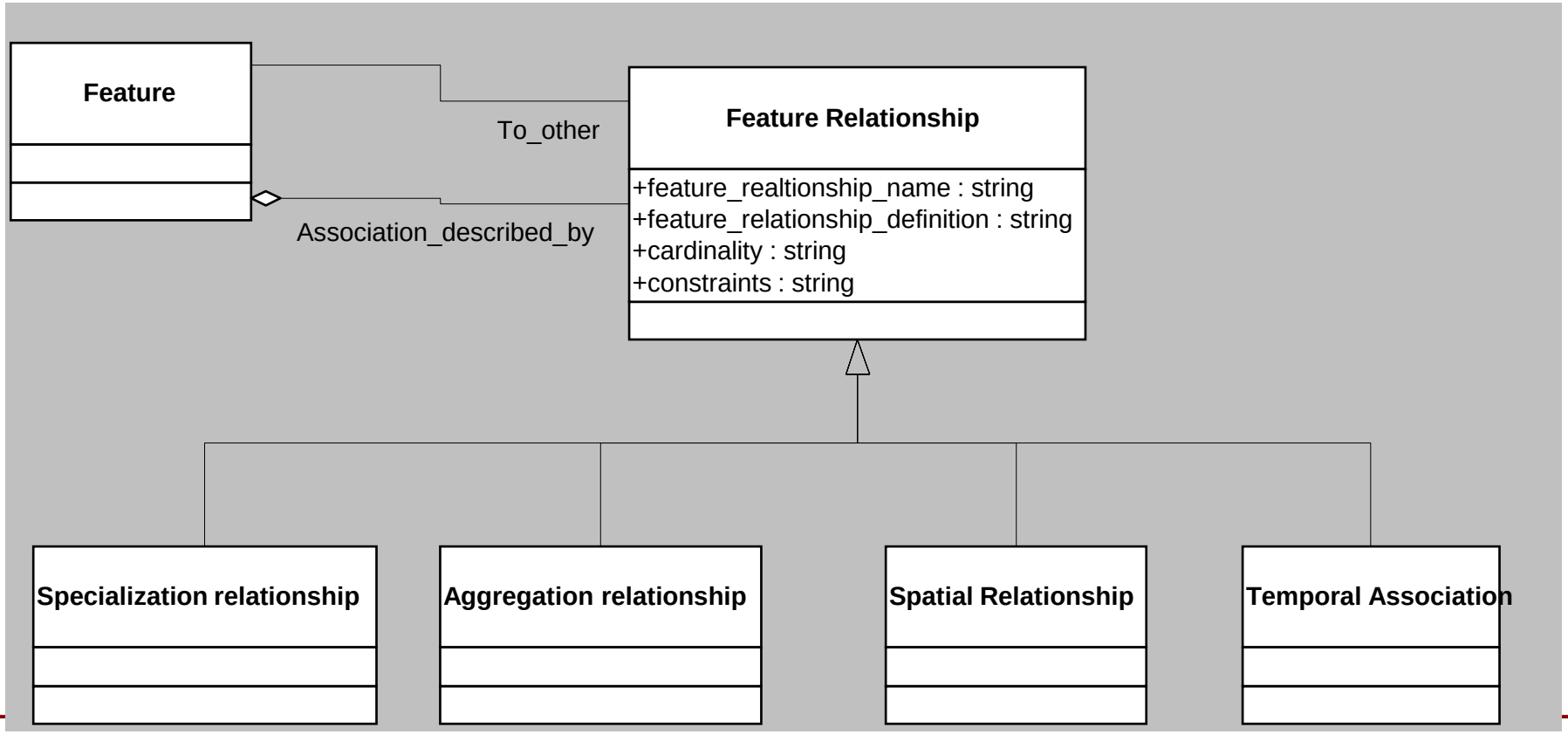
- To generate a data model adhering to the GFM you need to:
  - Identify these feature types
  - Give them names and definitions
  - Identify the properties (attributes) of these feature types
  - Decide on what form the values of these properties should take (the value of a property may be another feature or simple 'data types').
  - Identify the relationships between different feature types and the nature of those relationships
  - Identify any constraints that should be placed on the feature types
  - Encode all of the above in a UML model known as an 'Application Schema'.
  - This UML model then becomes the model – from which other forms (documentation, XML schemas, database schemas etc.) can be derived.

“The General Feature Model” – Details (important)



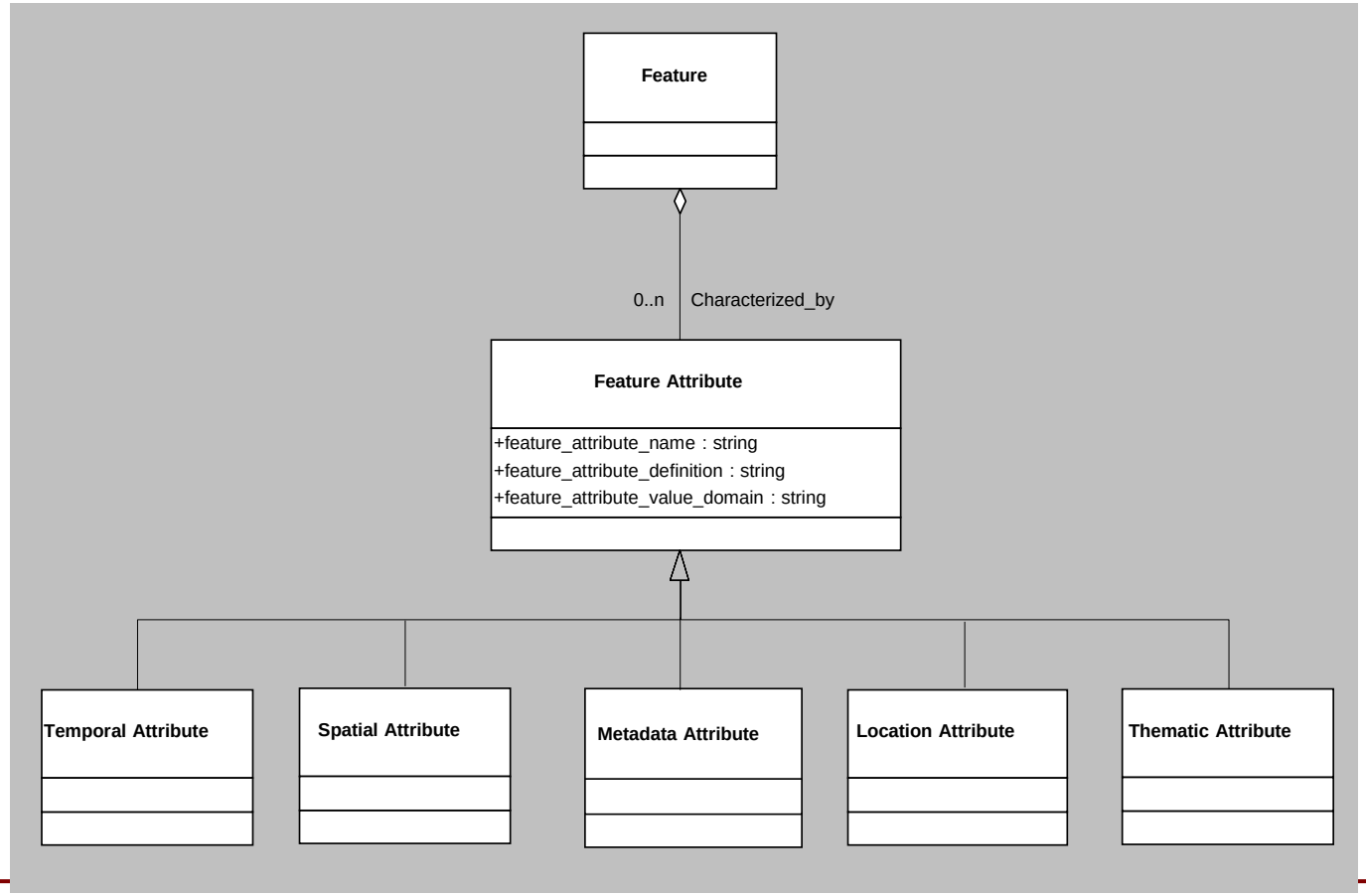
“The General Feature Model” relationship description (important)

- Types of feature relationships that can be used to describe features



“The General Feature Model” attribute description (important)

- Types of attributes that can be used to describe features



Useful Links

- Working with INSPIRE Models in Enterprise Architect

https://inspire.ec.europa.eu/portfolio/data-models

Note:

the actual version of EA has specialized structures for ArcGIS and GML data models

References – Abstract Models

- ISO 19109 Rules for application schema: contains the general feature model for ISO TC211. It guides the use of classes, relationships, interfaces, and properties in designing feature schemas for data transfers or transactions.

- ISO 19110 Feature cataloguing methodology: provides a basis for describing feature types to be pooled across a community of users.

- ISO 19107 Spatial Schema: provides a model of 2-dimensional and 3-dimensional geometry and topology, and related operators such as "buffer" or "intersects."

- ISO 19125-1 Simple Features Common Architecture: provides further detail on the subset of features described in OGC's Simple Feature Access Implementation Specifications, including well-known encodings and a starter set of Spatial Reference Systems.

- ISO 19108 Temporal Schema: defines how to represent features over time as well as in space.

- OGC Abstract Specifications: Topic 1 (Feature geometry), Topic 3 (Locational geometry), Topic 5 (The OpenGIS Feature), Topic 10 (Feature Collections).