

Australian/New Zealand Standard™

Geographic information—Rules for application schema



AS/NZS ISO 19109:2006

This Joint Australian/New Zealand Standard was prepared by Joint Technical Committee IT-004, Geographical Information/Geomatics. It was approved on behalf of the Council of Standards Australia on 11 January 2006 and on behalf of the Council of Standards New Zealand on 27 January 2006.
This Standard was published on 20 February 2006.

The following are represented on Committee IT-004:

ACT Planning and Land Authority
ANZLIC - the Spatial Information Council
Association of Crown Research Institutes New Zealand
Australasian Fire Authorities Council
Australian Antarctic Division
Australian Bureau of Statistics
Australian Hydrographic Office
Australian Map Circle
Australian Spatial Information Business Association
CSIRO Exploration & Mining
Department for Administrative and Information Services, SA
Department of Conservation, New Zealand
Department of Defence (Australia)
Department of Infrastructure, Planning and Environment, NT
Department of Natural Resources and Mines, Qld
Department of Primary Industries, Water and Environment, Tas
Geoscience Australia
Inter Governmental Committee on Surveying and Mapping
Land Information New Zealand
Land Victoria
Mapping Sciences Institute, Australia
Telecom New Zealand
Western Australian Land Information System

Keeping Standards up-to-date

Standards are living documents which reflect progress in science, technology and systems. To maintain their currency, all Standards are periodically reviewed, and new editions are published. Between editions, amendments may be issued. Standards may also be withdrawn. It is important that readers assure themselves they are using a current Standard, which should include any amendments which may have been published since the Standard was purchased.

Detailed information about joint Australian/New Zealand Standards can be found by visiting the Standards Web Shop at www.standards.com.au or Standards New Zealand web site at www.standards.co.nz and looking up the relevant Standard in the on-line catalogue.

Alternatively, both organizations publish an annual printed Catalogue with full details of all current Standards. For more frequent listings or notification of revisions, amendments and withdrawals, Standards Australia and Standards New Zealand offer a number of update options. For information about these services, users should contact their respective national Standards organization.

We also welcome suggestions for improvement in our Standards, and especially encourage readers to notify us immediately of any apparent inaccuracies or ambiguities. Please address your comments to the Chief Executive of either Standards Australia or Standards New Zealand at the address shown on the back cover.

This Standard was issued in draft form for comment as DR 05520.

Australian/New Zealand Standard™

Geographic information—Rules for application schema

First published as AS/NZS ISO 19109:2006.

COPYRIGHT

© Standards Australia/Standards New Zealand

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of the publisher.

Jointly published by Standards Australia, GPO Box 476, Sydney, NSW 2001 and Standards New Zealand, Private Bag 2439, Wellington 6020

ISBN 0 7337 7254 4

PREFACE

This Standard was prepared by the Joint Standards Australia/Standards New Zealand Committee IT-004, Geographical Information/Geomatics.

This Standard is identical with, and has been reproduced from ISO 19109:2005, *Geographic information—Rules for application schema*.

The objective of this Standard is to provide geographic information designers and analysts with defined rules for creating and documenting application schemas, including principles for the definition of geographic feature.

The terms ‘normative’ and ‘informative’ are used to define the application of the annex to which they apply. A normative annex is an integral part of a standard, whereas an informative annex is only for information and guidance.

As this Standard is reproduced from an international standard, the following applies:

- (a) Its number appears on the cover and title page while the international standard number appears only on the cover.
- (b) In the source text ‘this International Standard’ should read ‘this Australian/New Zealand Standard’.
- (c) A full point substitutes for a comma when referring to a decimal marker.

References to International Standards should be replaced by references to Australian or Australian/New Zealand Standards, as follows:

<i>Reference to International Standard</i>		<i>Australian/New Zealand Standard</i>	
ISO		ISO	
19107	Geographic information—Spatial schema	19107	Geographic information—Spatial schema
19108	Geographic information—Temporal schema	19108	Geographic information—Temporal schema
19112	Geographic information—Spatial referencing by geographic identifiers	19112	Geographic information—Spatial referencing by geographic identifiers
19113	Geographic information—Quality principles	19113	Geographic information—Quality principles
19115	Geographic information—Metadata	19115	Geographic information—Metadata
ISO/TS		ISO/TS	
19103	Geographic information—Conceptual schema language	19103	Geographic information—Conceptual schema language
19501	Information technology—Open Distributed Processing—Unified Modeling Language (UML) Version 1.4.2	—	

CONTENTS

	<i>Page</i>
1	Scope 1
2	Conformance 1
3	Normative references 2
4	Terms and definitions..... 2
5	Presentation and abbreviations 4
5.1	Presentation 4
5.2	Abbreviations 5
6	Context..... 5
6.1	Purpose of an application schema 5
6.2	Rules for application schema..... 5
6.3	Application schema supporting data interchange 6
7	Principles for defining features..... 8
7.1	Features 8
7.2	Features and the application schema..... 9
7.3	The General Feature Model..... 10
7.4	Attributes of feature types 16
7.5	Relationships between feature types 18
7.6	Behaviour of feature types 20
7.7	Constraints 21
8	Rules for application schema..... 21
8.1	The application modelling process..... 21
8.2	The application schema 22
8.3	Rules for application schema in UML..... 24
8.4	Domain profiles of standard schemas in UML 27
8.5	Rules for use of metadata schema 29
8.6	Temporal rules 33
8.7	Spatial rules..... 39
8.8	Cataloguing rules..... 49
8.9	Spatial referencing using geographic identifiers 50
Annex A (normative)	Abstract test suite..... 53
Annex B (normative)	The modelling approach and the General Feature Model 57
Annex C (informative)	Application schema in EXPRESS..... 61
Annex D (informative)	Application schema examples..... 63
Bibliography	 71

INTRODUCTION

Any description of reality is always an abstraction, always partial, and always just one of many possible “views”, depending on the application field.

The widespread application of computers and geographic information systems (GIS) has led to an increased use of geographic data within multiple disciplines. With current technology as an enabler, society’s reliance on such data is growing. Geographic datasets are increasingly being shared and exchanged. They are also used for purposes other than those for which they were produced.

To ensure that data will be understood by both computer systems and users, the data structures for data access and exchange must be fully documented. The interfaces between systems, therefore, need to be defined with respect to data and operations, using the methods standardized in this International Standard. For the construction of internal software and data storage within proprietary systems, any method may be used that enables the standardized interfaces to be supported.

An application schema provides the formal description of the data structure and content required by one or more applications. An application schema contains the descriptions of both geographic data and other related data. A fundamental concept of geographic data is the feature.

AUSTRALIAN/NEW ZEALAND STANDARD

Geographic information — Rules for application schema**1 Scope**

This International Standard defines rules for creating and documenting application schemas, including principles for the definition of features.

The scope of this International Standard includes the following:

- 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.

The following are outside the scope:

- choice of one particular conceptual schema language for application schemas;
- definition of any particular application schema;
- representation of feature types and their properties in a feature catalogue;
- representation of metadata;
- rules for mapping one application schema to another;
- implementation of the application schema in a computer environment;
- computer system and application software design;
- programming.

2 Conformance

Any application schema claiming conformance to this International Standard shall pass all of the requirements described in the abstract test suites in Annex A.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/TS 19103:—¹⁾, *Geographic information — Conceptual schema language*

ISO 19107:2003, *Geographic information — Spatial schema*

ISO 19108:2002, *Geographic information — Temporal schema*

ISO 19112:2003, *Geographic information — Spatial referencing by geographic identifiers*

ISO 19113:2002, *Geographic information — Quality principles*

ISO 19115:2003, *Geographic information — Metadata*

ISO/IEC 19501, *Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2*

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

4.1

application

manipulation and processing of data in support of user requirements

[ISO 19101]

4.2

application schema

conceptual schema for data required by one or more applications

[ISO 19101]

4.3

complex feature

feature composed of other features

4.4

conceptual model

model that defines concepts of a universe of discourse

[ISO 19101]

4.5

conceptual schema

formal description of a conceptual model

[ISO 19101]

1) To be published.

4.6**dataset**

identifiable collection of data

[ISO 19115]

4.7**domain**

well-defined set

[ISO 19107]

NOTE Well-defined means that the definition is both necessary and sufficient, as everything that satisfies the definition is in the set and everything that does not satisfy the definition is necessarily outside the set.

4.8**feature**

abstraction of real-world phenomena

NOTE A feature may occur as a type or an instance. Feature type or feature instance should be used when only one is meant.

[ISO 19101]

4.9**feature association**

relationship that links instances of one feature type with instances of the same or a different feature type

[ISO 19110]

4.10**feature attribute**

characteristic of a feature

NOTE 1 A feature attribute may occur as a type or an instance. Feature attribute type or feature attribute instance is used when only one is meant.

NOTE 2 A feature attribute type has a name, a data type and a domain associated to it. A feature attribute instance has an attribute value taken from the domain of the feature attribute type.

[adapted from ISO 19101]

4.11**feature operation**

operation that every instance of a feature type may perform

EXAMPLE 1 A feature operation upon the feature type “dam” is to raise the dam. The results of this operation are to raise the height of the “dam” and the level of water in a “reservoir”.

EXAMPLE 2 A feature operation by the feature type “dam” might be to block vessels from navigating along a watercourse.

[adapted from ISO 19110]

4.12**geographic data**

data with implicit or explicit reference to a location relative to the earth

NOTE Geographic information is also used as a term for information concerning phenomena implicitly or explicitly associated with a location relative to the earth.

4.13**metadata**

data about data

[ISO 19115]

4.14**model**

abstraction of some aspects of reality

4.15**portrayal**

presentation of information to humans

[ISO 19117]

4.16**quality**

totality of characteristics of a product that bear on its ability to satisfy stated and implied needs

[ISO 19101]

4.17**universe of discourse**

view of the real or hypothetical world that includes everything of interest

[ISO 19101]

5 Presentation and abbreviations

5.1 Presentation

This International Standard describes how to create an application schema that integrates conceptual schemas defined in the ISO 19100 series of International Standards for geographic information. In addition to stating the rules for creating application schemas, this International Standard provides guidance through examples. This International Standard adopts the following conventions for presentation purposes:

a) Rules:

All rules are normative, and are described as follows.

Rules:

1) <Rule 1>

2) <Rule 2>

b) Tables:

Tables that are not referenced from the rules are informative.

c) Conceptual schemas:

Conceptual schemas in the normative part of this International Standard are presented in the Unified Modelling Language (UML) in conformance with ISO/TS 19103. UML diagrams are presented in compliance with ISO/IEC 19501.

5.2 Abbreviations

CSL	Conceptual schema language
GFM	General feature model
OCL	Object constraint language
UML	Unified modelling language

6 Context

6.1 Purpose of an application schema

An application schema is a conceptual schema for data required by one or more applications. An application schema defines

- content and structure of data; and
- specifications of operations for manipulating and processing data by an application.

The purpose of an application schema is twofold:

- to provide a computer-readable data description defining the data structure, which makes it possible to apply automated mechanisms for data management; and
- to achieve a common and correct understanding of the data, by documenting the data content of the particular application field, thereby making it possible to unambiguously retrieve information from the data.

6.2 Rules for application schema

This International Standard does not standardize application schemas; it only defines rules for creating application schemas in a consistent manner (including the consistent definition of features) to facilitate the acquiring, processing, analysing, accessing, presenting and transferring of geographic data between different users, systems and locations. The rules in this International Standard are, in the case of data transfer or interchange, used by suppliers and users of geographic data to

- build a transfer application schema for data interchange;
- interpret the semantics of the transferred dataset with respect to user's local data and content and structure of data; and
- determine the necessary transformations between the two datasets.

The rules in this International Standard will assist the users of applications with similar data requirements in creating a common application schema for the interface between their systems and data. This includes an agreement about the elements from the universe of discourse. This is described in more detail in 6.3.

The mapping from one application schema to another application schema may be difficult and even impossible if the two schemas are too divergent. The mapping is facilitated if the application schema used within a system is designed considering also the requirements for the data interchange. The rules can also be used for building an application schema used within a system, although such application schemas are not within the scope of this International Standard.

The creation of an application schema is a process. The content of an application schema has to be settled according to the view of reality in the universe of discourse. This is modelled in terms of types of features and their properties. Clause 7 contains principles for consistently defining features.

The application schema defines the structure and content of data. It is expressed in a conceptual schema language (CSL). Clause 7 also includes a model expressed in UML that defines the concepts required to describe types of features. Feature type definition may be documented in feature catalogues. Such definitions may be used in an applications schema. Other standards in the ISO 19100 series define reusable modules of conceptual schemas that may be integrated in an application schema. Clause 8 gives the main rules for integrating these predefined modules into a conceptual schema in UML.

NOTE ISO 19118 defines how a dataset defined by an application schema in UML is encoded.

6.3 Application schema supporting data interchange

6.3.1 Introduction

Data interchange between information systems may take place in two ways.

- In the traditional data transfer model, the data supplier creates a dataset that is transferred to the user. The structure and the content of data are described in the application schema for the dataset. The dataset is sent in a transfer format.
- In the interoperability model, the user application communicates with the supplier application through a common communication protocol. In this scenario, the user invokes services that result in data being passed from the service provider to the user application. The application schema describes not only the structure and content of the exchanged data, but also the structure of the interfaces involved in the transaction.

There is a fundamental distinction between a data transfer and a data transaction. In data transfer, a data set is predefined in an application schema. The spatial extent and the rules for inclusion of feature instances are also predefined. The user requests and receives a copy of the dataset (or may receive it automatically through a long-term agreement for distribution of datasets). In a data transaction, a requester first specifies selection criteria, such as spatial extent and feature instance inclusion rules for the data from the producer's data store. Data meeting the selection criteria are then retrieved from the data store and provided to the user.

NOTE Conformance to the rules in this International Standard does not guarantee that data conforming to any given application schema can be transformed in a meaningful way to conform to any other application schema. At best, it allows the user to determine which elements are common to the two schemas and which could be transformed from one schema to another, as well as those that cannot be transformed. Complete interoperability can only occur when user and supplier have identical application schemas.

6.3.2 Data interchange by transfer

Figure 1 shows the traditional data transfer model for data suppliers and data users. The structure and content of the data provided by the supplier and received by the user are described in application schemas. To be able to transfer data, three conditions must be fulfilled.

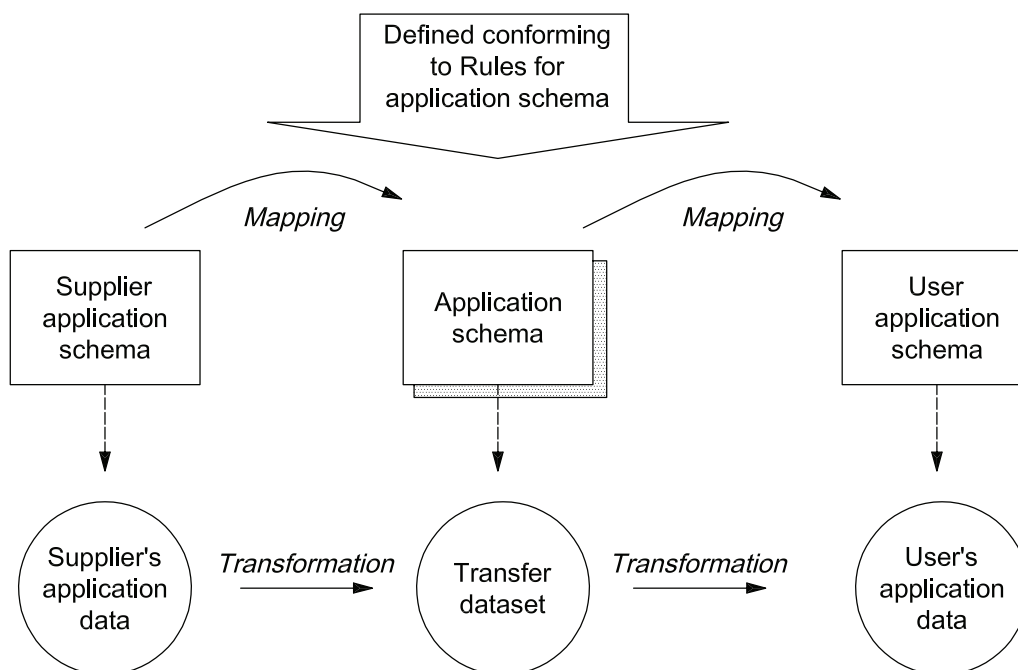


Figure 1 — Data interchange by transfer

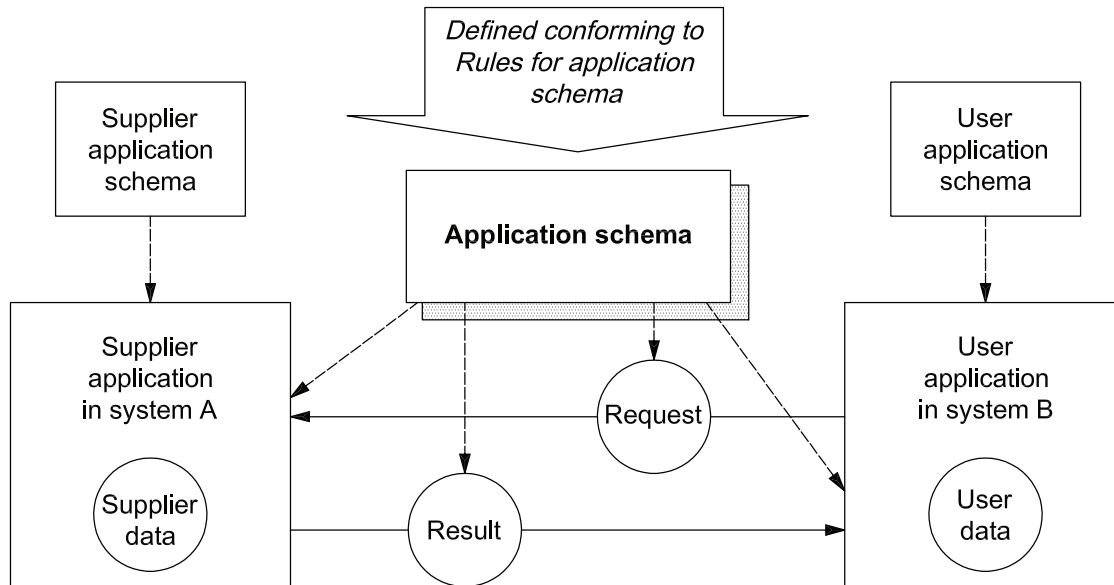
First, the user and the supplier must agree on creating an application schema for the data being exchanged in accordance with this International Standard. In order to facilitate the transfer of data, this application schema shall be developed using the application schemas from the user and the supplier. One mapping will be made from the supplier's application schema to this application schema for the exchanged data, and a second mapping will be made from this application schema to the application schema of the user.

Second, the supplier must be able to transform the application data defined according to the supplier application schema into a transfer dataset defined according to the application schema for the exchanged data.

Third, the user must be able to transform the transfer dataset defined according to its application schema to the application data defined according to the user's application schema.

6.3.3 Data interchange by transactions

Figure 2 shows data interchange through transactions described in the interoperability model. The user application makes a request for data that is received by the supplier application. In response, the supplier application delivers a resulting dataset. Both the request and the resulting dataset are defined according to a common application schema. The supplier application is responsible for transforming the data in system A into the data in the exchanged dataset. After receipt, the user application is responsible for transforming the exchanged data into data in system B. Data interchange by transaction is provided by geospatial services as defined in ISO 19119. In particular, feature access services are defined in Geographic model/information management services.



NOTE The unbroken lines show the flow of data. Broken lines denote the role of the application schema on the data interchange.

Figure 2 — Data interchange by transactions

7 Principles for defining features

7.1 Features

A fundamental unit of geographic information is called a feature. ISO 19110 provides a standard framework for organizing and reporting the classification of features. It also gives a broader discussion on different aspects of geographic features.

This International Standard gives rules for creating application schemas, including the principles for the definition of features. The term feature is used in different contexts defined according to the four-layer architecture. Annex B describes the use of the term feature in the four-layer architecture.

This International Standard distinguishes four aspects of defining features: the definitions or description used to group them into types, the attributes associated with each type, the relationships among the types and the behaviour of the features.

EXAMPLE “Tower Bridge” is the abstraction of a certain real-world bridge in London. The term “bridge” is the abstraction of the collection of all real-world phenomena that is classified into the concept behind the term “bridge”. Later in the document, the terms “feature type” and “feature instance” are used to separate the concept of “feature” describing the whole collection from the concept of describing a certain instance occurrence.

Figure 3 describes the most abstract level of defining and structuring geographic data. The classification of real-world phenomena as features depends on their significance to a particular universe of discourse.

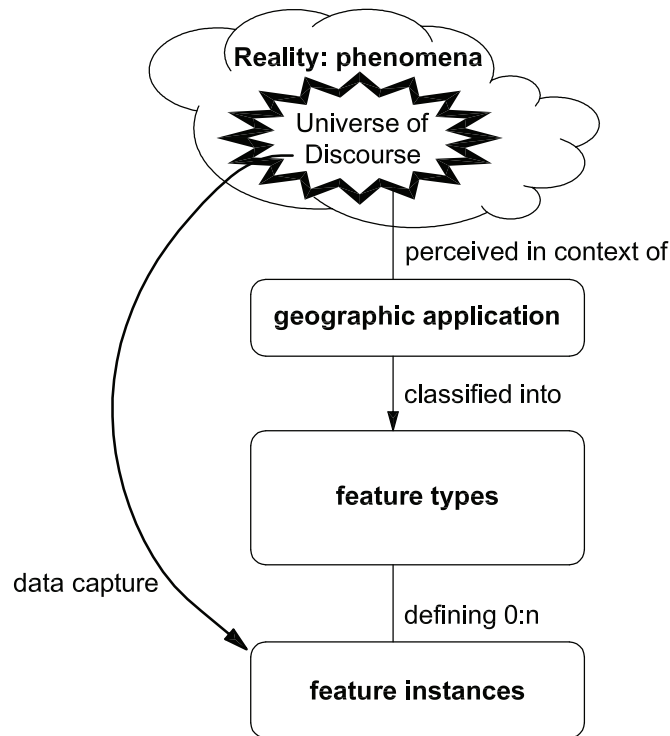


Figure 3 — The process from universe of discourse to data

7.2 Features and the application schema

This International Standard supports the definition of features with respect to their representation in data structures defined by application schemas.

Figure 4 shows the process of structuring data from the universe of discourse to the geographic dataset. The definitions of the feature types and their properties, as perceived in context of an application field, will be derived from the universe of discourse. A feature catalogue documents the feature types.

An application schema defines the logical structure of data and may define operations that can be performed on or with the data. An application schema addresses the logical organization, rather than the physical.

The developer of an application schema may use feature definitions from feature catalogues that already exist. This will reduce the costs of data acquisition, allowing the developer to use existing data, and simplify the process of developing the application schema.

The application schema shall be expressed in a conceptual schema language. Each conceptual schema language has its own terms and concepts. When creating an application schema, the concepts of the General Feature Model (GFM; see 7.3) are mapped to the concepts of the chosen conceptual schema language. For UML, these rules are described in 8.3.

NOTE Annex C gives as an example the rules for mapping the concepts of the GFM to the concepts of ISO 10303-11.

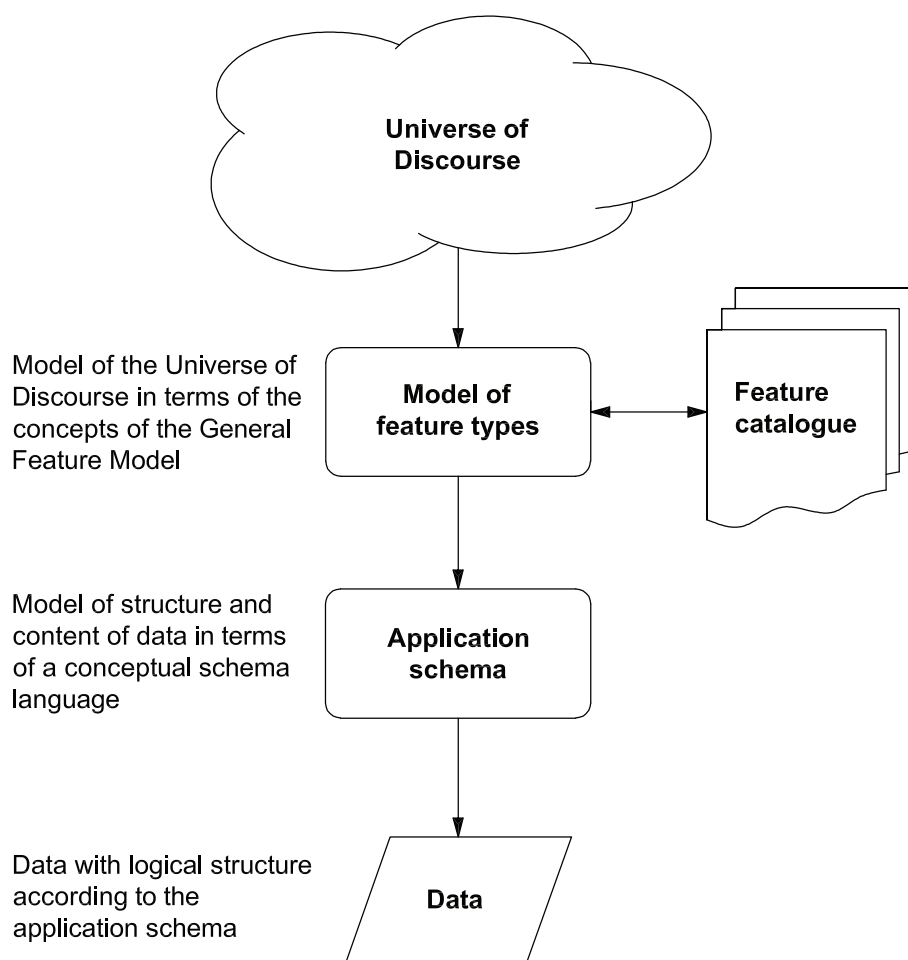


Figure 4 — From reality to geographic data

7.3 The General Feature Model

7.3.1 Introduction

This subclause identifies and describes the concepts used to define features and how these concepts are related. The description is expressed in a conceptual model, also called the General Feature Model (GFM).

Annex B provides discussions regarding the purpose and design of the GFM.

NOTE The complete kernel of GFM is found in Annex B.3.

The concepts of the GFM are used in the feature catalogue structure described in ISO 19110. ISO 19117 also uses these concepts for specifying the portrayal of geographic information. The concepts of the GFM are used to establish categories of geographic processing services taxonomy in ISO 19119.

Subclauses 7.4 to 7.7 depict how different aspects of the properties of a feature are managed. Subclause 7.4 describes different aspects of attributes and 7.5 describes different aspects of relationships. Subclause 7.6 includes a more detailed description of behaviour of features, and 7.7 describes the concept of constraints.

7.3.2 The purpose of the GFM

The GFM is a model of the concepts required to classify a view of the real world. It is expressed in a CSL, 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.

The things we want to classify we call features; the relations between feature types are feature association types and inheritance. Feature types have properties that are feature attributes, feature operations and feature association roles. All these concepts are expressed as UML metaclasses in the GFM. The GFM is a metamodel of feature types.

Feature types may be documented in feature catalogues. The GFM may serve as the conceptual model of the feature-catalogue structure, but the feature catalogue has additional concepts for documenting feature types. There is a Feature Catalogue Model (FCM) that realizes the GFM concepts and also adds some more concepts (see ISO 19110). Some new concepts are, for example, a list of the feature attributes of each feature type, aliases of the feature type name, and codes for the feature type name. These additional concepts are not in conflict with the concepts from the GFM.

An application schema shall be expressed in a CSL. It shall describe the structure and content of the dataset that represents a universe of discourse. The GFM specifies the requirements for the classification of features, but is not a CSL. This means that we have to use an existing CSL to define the application schema. Within the ISO 19100 series of International Standards, UML is used. As we want to integrate standard schemas from the ISO 19100 series into our application schemas, it is convenient to express the application schema in UML. In this International Standard, we have defined the main rules for mapping the GFM concepts into the UML language. This may be done for other CSLs as well; in Annex C we have exemplified the mapping from GFM to ISO 10303-11 EXPRESS.

The GFM defines the structure for classifying features that we need to keep in mind when we make our application schema in UML. However, the mapping from GFM to UML is a one-way mapping; it is not possible to map backwards. For example, the application schema has UML classes. Some of these classes are GFM feature types and some are the datatypes for feature attributes; it is not possible to keep these things apart. The GFM does not define feature-attribute values to the depth that is needed. That is not necessary to do as the GFM only specifies the structure and content of definition of features.

The conclusion is that the GFM is a metamodel for definition of features that also is used to define the structure of feature catalogues. The chosen CSL metamodel (e.g. the UML metamodel as restricted by ISO 19103) is the metamodel for an application schema. As the application schema deals with data representing features, the structure of the GFM has to be kept in mind while creating the application schema.

7.3.3 The main structure of the GFM

Figure 5 shows the concepts used to define types of features. Figure 5 is an extract from the whole model. Subclause B.3 provides a figure (Figure B.2) showing all concepts of the GFM and the relationships between them.

Besides a name and a description, a feature type is defined by its properties such as

- feature attributes;
- feature association roles characterizing the feature type; and
- defined behaviour of the feature type.

Additional concepts are

- feature associations between the feature type and itself or other feature types;
- generalization and specialization relationships to other feature types; and
- constraints on the feature type.

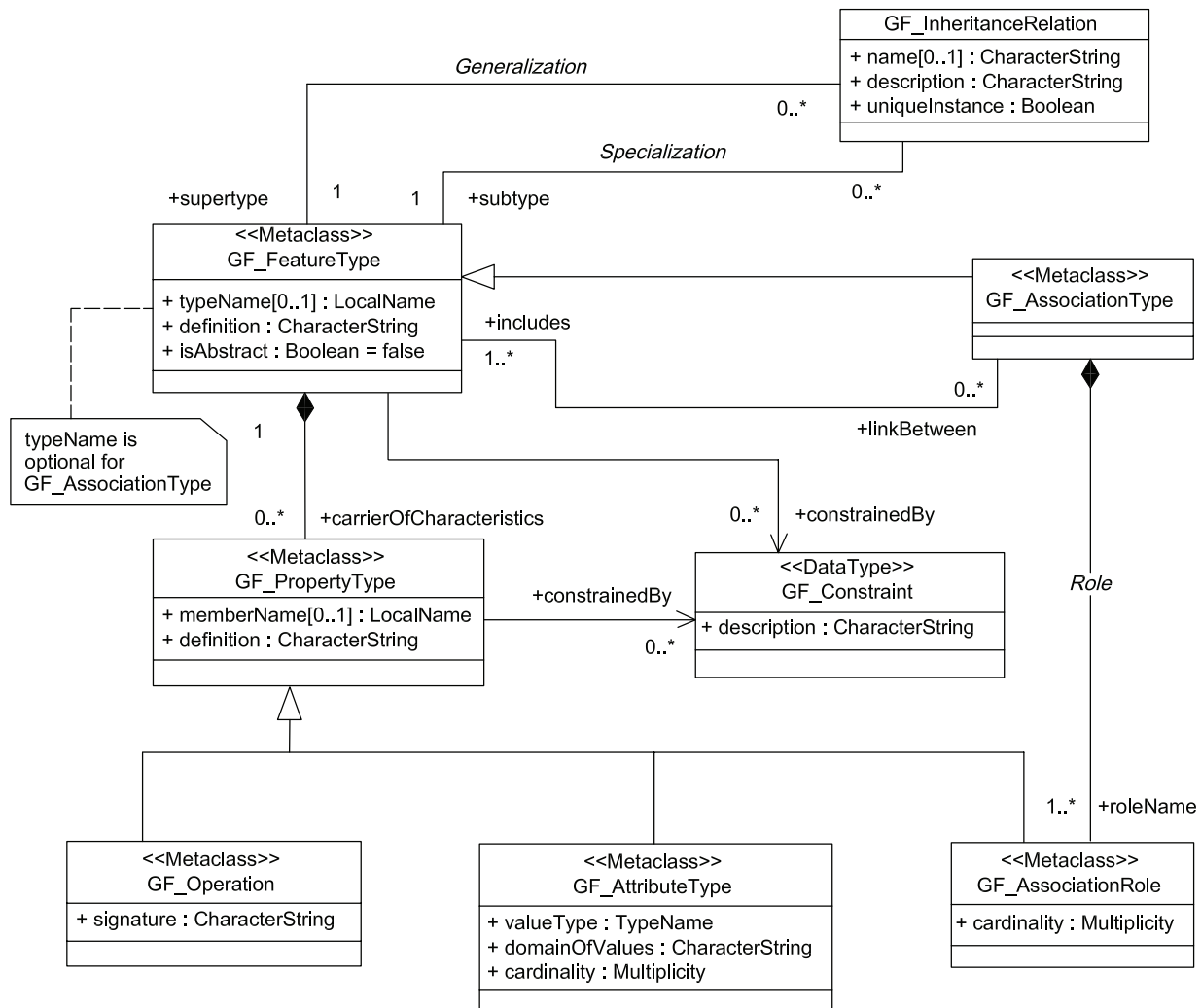


Figure 5 — Extract from the General Feature Model

NOTE Figure 8 shows additional relationships from GF_Operation to GF_AttributeType and GF_AssociationType.

7.3.4 GF_FeatureType

A feature is an abstraction of real-world phenomena. GF_FeatureType is a metaclass that is instantiated as classes that represent individual feature types. A certain feature type is the class for all instances of that feature type. The instances of a class that represents an individual feature type are feature instances.

NOTE 1 Feature types are equivalent to classes and feature instances are equivalent to objects, in object-oriented modelling.

NOTE 2 Annex B provides a table on the use of the term “feature”.

— typeName

name of the feature type. The name shall be unique within an application schema. TypeName is optional only for GF_AssociationType.

— LocalName

identifier within a name space for a local object. Types defined in ISO/TS 19103, LocalName is a subtype of GenericName which is a component of the NameSpace definition. This could be the target object of

the GenericName, or a pointer to another NameSpace (with a new GenericName) one step closer to the target of the identifier.

— **definition**

definition that describes the feature type.

— **isAbstract**

Boolean attribute. If true, the feature type acts as an abstract supertype.

— **includes**

The association role specifies that an instance of a feature association may include any number of instances of feature types.

7.3.5 GF_PropertyType

GF_PropertyType is the metaclass for any class of property of a feature type that describes characteristics of the feature, the behaviour of a feature, or the association roles that the feature is in.

GF_Property Type is the supertype of GF_Operation, GF_AttributeType and GF_AssociationRole.

— **memberName**

name of the behaviour, attribute or role. Only name of role is optional.

— **LocalName**

identifier within a name space for a local object. This could be the target object of the GenericName, or a pointer to another NameSpace (with a new GenericName) one step closer to the target of the identifier. In the Basic Types defined in ISO/TS 19103, LocalName is a subtype of GenericName which is a component of the NameSpace definition.

— **definition**

— description of the behaviour, attribute or role of a feature type.

— **carrierOfCharacteristics**

The association role carrierOfCharacteristics specifies that any feature operation, any feature attribute type and any feature association role carries characteristics of a feature type.

7.3.6 GF_AttributeType

— GF_AttributeType is the metaclass for attribute definitions of a feature type (see also 7.4).

— **valueType**

data type of the attribute value.

NOTE ISO/TS 19103 defines data types that may be used for the valueType of a feature attribute.

EXAMPLE 1 Integer, character String or GM_Object.

— **TypeName**

identifier within a type space for a local object. In the Basic Types defined in ISO/TS 19103, TypeName is subtype of LocalName (see 7.3.4).

— **domainOfValues**

description of a set of values.

EXAMPLE 2 Positive, from 3 to 7, GM_Object and all its subtypes as defined in ISO 19107.

— **cardinality**

number of instances of the attribute that may be associated with a single instance of a feature type.

— **Multiplicity**

specification of the range of allowable cardinalities that a set may assume. This data type is defined in ISO/TS 19103.

7.3.7 GF_AssociationRole

GF_AssociationRole is the metaclass for the classes of roles that are part of a GF_AssociationType (see 7.3.9).

NOTE GF_AssociationRole indicates the role played by the feature type through the association. The instance of GF_AssociationRole that gives the role for one feature type can also be seen as part of this feature type.

— **cardinality**

number of instances of the feature type that can act in this role relative to a single instance of the feature type at the other end of the association.

— **Multiplicity**

specification of the range of allowable cardinalities that a set may assume.

— **Role**

The association role specifies the roles associated to a GF_AssociationType.

— **roleName**

role of being a specific role assigned to a GF_AssociationType.

7.3.8 GF_Operation

GF_Operation is the metaclass for describing behaviour of feature types in terms of operations (see also 7.6).

NOTE 1 GF_operations only apply to the interoperability model and do not apply to the data transfer model as described in 6.3.1

NOTE 2 Instances of GF_Operation are of three kinds: observer operations, mutator operations and constructor operations. Observer operations return the current values of attributes. Mutator operations include actions that change those values. A mutator operation creates an instance of a class for which it is defined. For example, an observer operation may be used to find the height of a dam. Raising the dam is a mutator operation that changes the height of the dam and also affects the attributes of the watercourse and the reservoir associated with the dam. Values may be observed or affected for another feature instance if there is an association between the feature types involved.

— **signature**

description that indicates the name, the arguments and the return values of an operation.

NOTE 3 In UML, signature is expressed in the form *operation_name(input_parameter1, input_parameter2,...) : output_value_type*, for example *has_height() : real*.

7.3.9 GF_AssociationType

GF_AssociationType is the metaclass for describing associations between feature types (see also 7.5). A feature association may have attributes. This is allowed because GF_AssociationType is a subtype of GF_FeatureType.

NOTE GF_AssociationType is subclassed under GF_Feature type for a variety of reasons. First, from a philosophical point of view, an association instance between feature instances meets the requirements in the definition of feature as an “abstraction of real-world phenomenon”. In the case of an instance of an association type, the “phenomenon” is the interaction of the two features. Second, from a pragmatic point of view, because an association meets the philosophical test for feature, it also meets some of the more pragmatic requirements. Associations often have spatial attributes, such as the location of the feature interaction. Associations may need to carry other attributes to describe the interaction, such as in 2D, road-rail intersections need to be classified as “road-overpass”, “road-underpass”, “rail-overpass”, “rail underpass” or “at-grade”, and may need to carry other attributes such as “clearance”. In many cases, the infrastructure at the point of interaction is naturally treated as a feature in its own right (such as a “rail-bridge”). In addition to the classical feature attributes, association instances can also carry metadata information. Pragmatically, association instances carry attributes just like feature instances, and inheriting the semantics and the code to accomplish that simplifies geographic information software.

— linkBetween

The association role linkBetween specifies that a GF_AssociationType will be a link from one instance of a feature type to the same or another instance of a feature type.

7.3.10 GF_InheritanceRelation

GF_InheritanceRelation is the class for a generic relationship between a more general feature type (supertype) and one specialized feature type (subtype).

Any instance of a specialized feature type is also an instance of the general feature type.

EXAMPLE The feature type “bridge” may belong to both the general class of “transportation feature” for road features and to the general class of “hazards” for navigation features. A specific instance of “bridge” is then also an instance of “transportation feature” and “hazards”.

Each specialization expresses a purpose. A feature type can act as supertype in a number of generic relationships, each having a different purpose.

— name

name of generalization/specialization; optional.

— description

explanation of the generalization/specialization.

— uniqueInstance

UniqueInstance is a Boolean variable, where .TRUE. means that an instance of the supertype shall not be an instance of more than one of the subtypes, whereas .FALSE. means that an instance of the supertype may be an instance of more than one subtype.

— Generalization

The association Generalization specifies that a feature type has the role of being a supertype in an inheritance relationship with another feature type.

— **Specialization**

The association Specialization specifies that a feature type has the role of being a subtype in an inheritance relationship with another feature type.

— **supertype**

role of being the more generic feature type of one other or other feature types.

— **subtype**

role of being the more specific feature type of one other or other feature types.

7.3.11 GF_Constraint

GF_Constraint is the class for constraints that may be associated with feature types and the properties of feature types (see also 7.7).

— **description**

The constraint described in natural language and/or in formal notation.

— **constrainedBy**

The role specifies that constraint is made on the GF_FeatureType or specified on GF_Properties within a feature type.

7.4 Attributes of feature types

7.4.1 Introduction

This subclause describes in more detail the role of attributes of features. An attribute type (GF_AttributeType) has a name (memberName), a description (definition), a type, a domain and cardinality associated with it.

The attributes carry all static information of a feature. This covers both spatial and non-spatial properties. In the ISO 19100 series of International Standards, some attribute types are of specific interest. These types are shown in Figure 6 as subtypes of GF_AttributeType. The attribute provides the interface to these other ISO 19100 International Standards because it uses their schemas. The attribute type will get the value type definition from those schemas and the value domain according to those schemas. For example, a spatial attribute type (GF_SpatialAttributeType) will have its value type and value domain according to the definition of GM_Object or TP_Object described in ISO 19107.

7.4.2 attributeOfAttribute

The association attributeOfAttribute links an attribute to another attribute that describes some characteristics of the first attribute.

EXAMPLE An attribute that carries the position of a feature may have another attribute that holds the positional accuracy (data value of GF_QualityAttributeType) of this position.

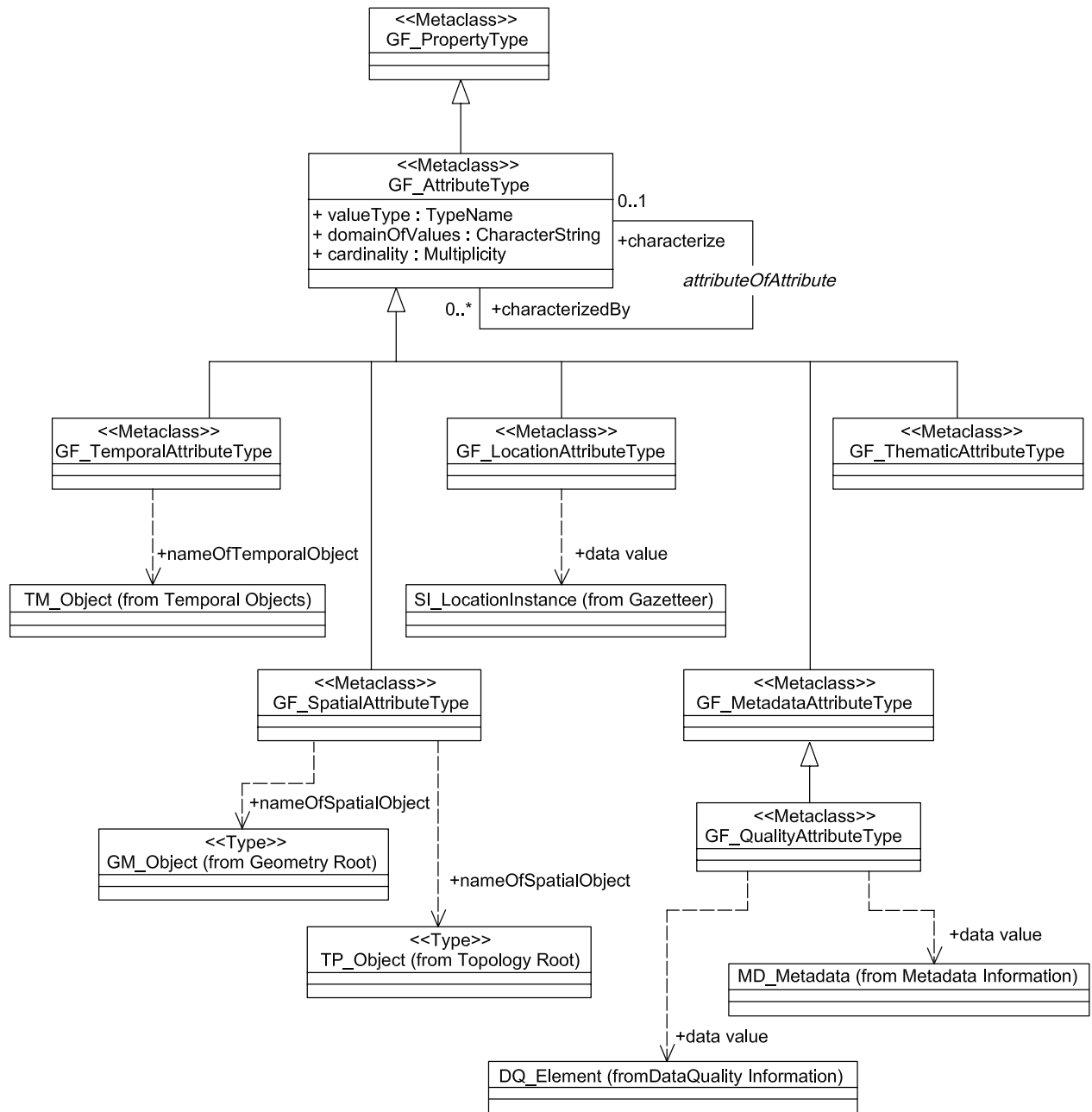


Figure 6 — Attributes of feature types

7.4.3 GF_SpatialAttributeType

GF_SpatialAttributeType represents a spatial attribute, which shall be used to express spatial characteristics of a feature type. A spatial attribute type shall have a GM_Object or a TP_Object as value type. The structures of GM_Object and TP_Object are defined in the Spatial Schema described in ISO 19107.

7.4.4 GF_TemporalAttributeType

GF_TemporalAttributeType represents a temporal attribute, which shall be used as the time reference characteristic of a feature. A temporal attribute type shall have a TM_Object as value type. The structure of TM_Object is defined in the Temporal Schema described in ISO 19108.

7.4.5 GF_QualityAttributeType

GF_QualityAttributeType represents attributes that carry quality information. These attributes shall be used when a quality characteristic of a feature or its properties is included as data in the dataset. Quality attribute shall have its value type according to the definition of DQ_Element defined in ISO 19115.

7.4.6 GF_LocationAttributeType

GF_LocationAttributeType represents attributes which carry a spatial reference of a feature by a geographic identifier.

EXAMPLE A postal area may be identified by a code, but its location can be found in a gazetteer.

These attribute types shall have their value type defined by SI_LocationInstance, which is defined in ISO 19112.

7.4.7 GF_MetadataAttributeType

GF_MetadataAttributeType represents attributes that carry metadata information when such information is included as data in a dataset. These attribute types shall use the metadata element classes defined in the Metadata Schema (ISO 19115) as their value types.

7.4.8 GF_ThematicAttributeType

GF_ThematicAttributeType represents attributes which carry any other descriptive characteristic of a feature except those specified in 7.4.3 to 7.4.7. Their value types and value domains will normally be defined by the user or by the application area. Both basic types (see ISO/TS 19103) and user defined data types can be used.

NOTE Information about thematic attributes may be found in feature catalogues.

7.5 Relationships between feature types

7.5.1 Introduction

This subclause describes relationships between feature types in some more detail. Figure 7 shows that relationships are classified as follows:

- generalization/specialization of feature types; and
- associations between features.

NOTE Association is a kind of relationship that may exist between both feature types and instances of feature types. The generalization/specialization relationship only occurs between feature types.

7.5.2 GF_InheritanceRelation

The GF_InheritanceRelation represents the specialization and generalization that specifies subtypes and supertypes of feature types. These relationships exist only between feature types, not between feature instances. A typical and powerful property of this kind of relationship is that a subtype will inherit all properties of its supertype.

EXAMPLE A generalization/specialization relationship between the feature types “lake” and “water body” says that an instance of the type “lake” is also an instance of the type “water body”; there are two types but only one instance involved in the relationship.

7.5.3 GF_AssociationType

GF_AssociationType represents all other association types between feature types. These relationships will appear both as association types when they are defined and as instances in the dataset. GF_AssociationType

is modelled as a subtype of `GF_FeatureType`. Accordingly, by definition of feature types in GFM, an association type may be characterized by its own properties, for example having its own attributes.

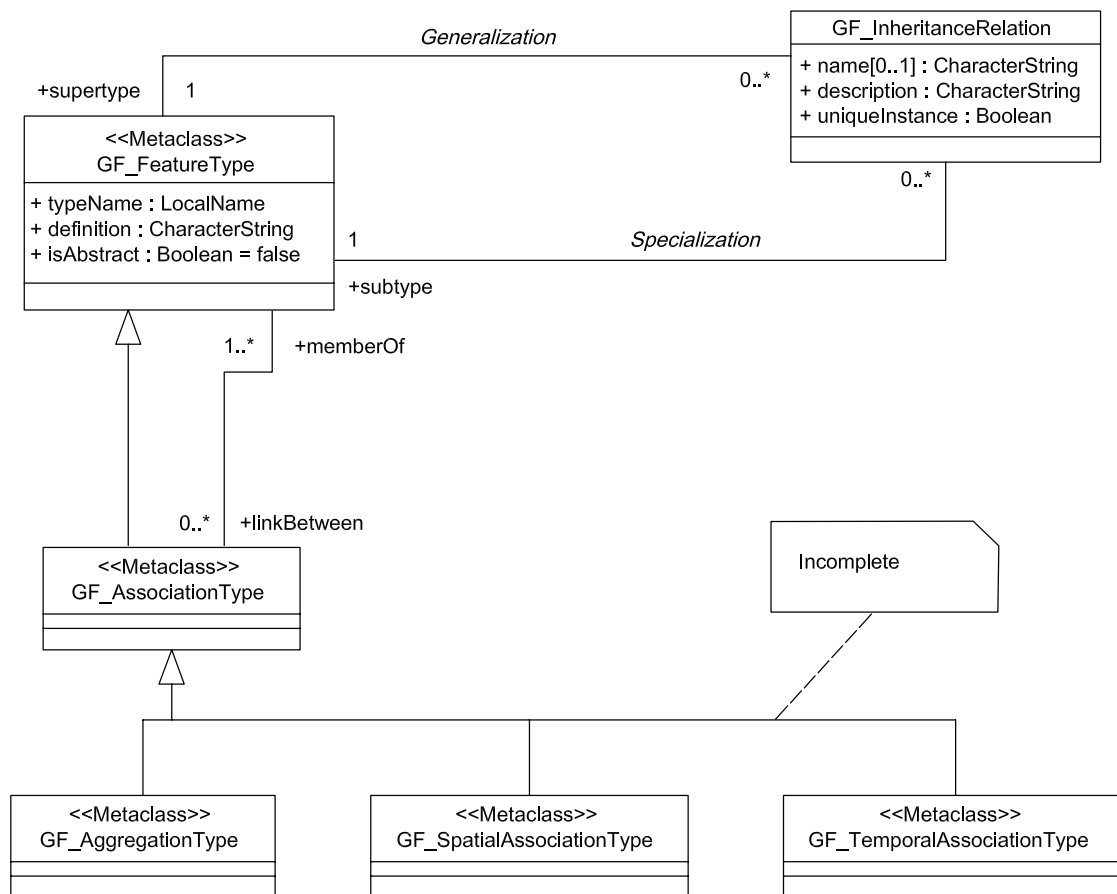


Figure 7 — Relationships between feature types

In the area of geographic information, there exist multiple kinds of associations, which shall be handled by rules defined in this International Standard. Three of these are shown in Figure 7 as subtypes of `GF_AssociationType`.

— **GF_AggregationType**

`GF_AggregationType` represents associations between the feature types that are some kind of “whole-part” relationship. The concept of aggregation includes both aggregation where the parts can exist even if the aggregate is destroyed and aggregation where the parts live inside the aggregate and will be destroyed together with the aggregate. Aggregation shall be used to specify feature types which form a complex feature type.

— **GF_SpatialAssociationType**

`GF_SpatialAssociationType` represents spatial relationships or topological relationships that may exist between features. Those relationships depend upon the relative spatial position of features. See further discussion in 8.7.4.

EXAMPLE 1 Sewerage network consisting of sewers and manholes.

EXAMPLE 2 An intersection is a relationship between two roads, and a road is a relationship between two intersections.