Working Committee of the Surveying Authorities
of the States of the Federal Republic of Germany (AdV)

Documentation on the Modelling of Geoinformation of Official Surveying and Mapping in Germany

(GeoInfoDok)

- Version 3.0 -

Bonn, April 1, 2004

Edited by
the AdV-working groups
Geo-Topography
Information and Communication Technology
Real Estate Cadastre
Spatial Referencing

Table of contents

1 Structure, content and objective

1.1 Initial situation, motive and objectives

The task of the surveying, mapping, and cadastral authorities of the federal states of Germany is to provide fundamental data for spatial referencing (Geobasis Data) for the use of official, industrial and private users. The demand for this data to be provided in digital format continues to increase and has been met at a very early stage by the authorities, which up to now record and provide the data of the real estate cadastre in the ALK (Automated Real Estate Map) and ALB (Automated Real Estate Register) and the topographic data in the ATKIS (Official Topographic Cartographic Information System) in a digital, standardized manner across the whole of Germany. Most Federal States are governed by a cabinet ruling that ALK and ATKIS data shall be used as a basis for other technical information systems (FIS).

The concepts according to which ALB, ALK and ATKIS were founded originated in the 1970s and 1980s. These concepts remain the platform on which the relevant geobasis data inventories are created and maintained at the moment. Other extensive digital database inventories have also been created according to the states’ specific concepts, e.g. digital orthophotos, raster data of the topographical state maps and digital elevation models.

In light of the rapidly developing technology, the growing wealth of experience gained by manufacturers in data recording and the changing requirements on the side of the users arising from utilisation of such data, it has become necessary to examine and further develop these concepts.

The existing information systems ALK and ALB will therefore in the future be integrated into the information system **ALKIS** (Official Real Estate Cadastre Information System). A harmonisation process in respect of the data model, the content and the semantics has also been carried out in line with ATKIS.

The **Digital Terrain Models** (DGM) are no longer a specific object group within the digital landscape model of ATKIS, but are now defined as a separate component. Similar to the control point objects of basic surveying, the universality of the DGM as an independent database is clarified and the opportunity to create combined data inventories or products using data from other product groups is improved.

For **Digital Orthophotos** (DOP) there is an AdV-Standard, which is not currently understood to be a part of the common application schema, but which has nonetheless been incorporated into the overall documentation under the heading of *Photo-based data* in Chapter 2 *'The AFIS-ALKIS-ATKIS Reference Model and the Product Groups*.

Geoinformation of official surveying and mapping also includes information on the control stations. Because these originally belong neither to ALK nor to ATKIS, they are now modelled in their own information system called **Official Geodetic Control Station Information System** (AFIS) with a separate feature catalogue.

The AdV projects AFIS, ALKIS and ATKIS, with their nationally standardised features are described in a common form under the heading *Documentation for Modelling Geoinformation of Official Surveying and Mapping*. They are associated with each other in a **common reference model** and described in the following Chapters of this document as a **common application schema for AFIS, ALKIS and ATKIS.**

The common application schema provides for the recording and management of **metadata and quality data** in accordance with the ISO specifications.

1.2 Core data, feature catalogue and versioning

Core data are the data provided by all surveying authorities of the states of the Federal Republic of Germany in AFIS, ALKIS and ATKIS for all users throughout the country. This also includes the associated metadata. A subsequent expansion of the core database is to be expected.

The **feature catalogue** of the real estate cadastre and the topographic state survey have been semantically harmonised with a view to achieving a highly-standardised real world model. Harmonisation has benefits for both internal and external applications. It is based on the previous catalogues (specimen-OBAK, list of application types, ATKIS-OK).

A concept for versioning features is being introduced in connection with the description of the procedure for **user-specific updating of secondary databases** (NBA). States that use history management within the meaning of the stage solution defined by the AdV for ALKIS base their modelling and the functionalities of a history management precisely on this application schema expanded by the version concept. For ATKIS, a periodical storing of the whole data inventories is considered adequate and sufficient.

1.3 Target audience and users

For economic reasons superregional users and GIS industry are demanding nationally-standardised core data in terms of the content and structuring of the basic geodata. Taking a holistic view of official surveying and mapping, the core data inventories of AFIS, ALKIS and ATKIS should be merged to form a core database containing its geodata.

A fundamental procedure in the form of a stage concept is proposed for the **migration** from established inventories. The details of the migration concept shall be defined specifically for each state. A re-migration into the interfaces of the previous systems for an interim supply of data to the users would be practicable for a prolonged transitional period. The migration concept has only temporary significance and is therefore not included in the overall documentation.

2 The AFIS-ALKIS-ATKIS Reference Model

The task of the AFIS-ALKIS-ATKIS Reference Model is to put the data inventories described in this documentation and their associations in context. The objective is to

- identify components,
- simplify modularisation,
- show the correlation with existing standards and
- avoid double work within the components.

AFIS is the Official Control Station Information System and contains descriptive and illustrative data on the following product groups:

- AFIS data in primary database,
- digital AFIS extracts and
- analogue AFIS extracts.

ALKIS is the Official Real Estate Cadastre Information System and contains real estate descriptive and illustrative data on the following product groups:

- ALKIS data in primary database,
- digital ALKIS extracts and
- analogue ALKIS extracts.

ATKIS is the Official Topographic and Cartographic Information System of the German state survey. ATKIS describes the landscape with various application objectives in the following product groups:

- digital landscape models (ATKIS-DLM)
- digital terrain models (DGM),
- digital topographic maps (DTK)
- analogue extracts from the digital topographic maps (DTK) and
- digital image models (DBM) in the form of digital orthophotos (DOP).

The content, structures and production instructions for the products of the reference model are defined at **control level** through the feature catalogue (OK) and signature catalogue (SK). These include:

- instructions for the structure of the information on fixed points, real estate cadastre and topography,
- instructions for creating presentation and map geometry objects (additional data),

- instructions for the presentation of objects,
- instructions for the form of analogue extracts.

The recording templates at **production level** are sub-divided into landscape models, digital image models (digital orthophotos) as well as maps and other documents. The landscape is used as the source of original information especially within the context of maintenance as a recording source. The digital data flow moves registered data into the primary database data of AFIS, ALKIS and ATKIS either directly without detouring via analogue media or after structuring and classification. The created geobasis data inventories can be re-used again immediately as a recording source for derived data inventories, e.g. components of the ALKIS primary database data, specifically building data are the basis for deriving associated data for the ATKIS-DLM. The recording operation includes the formation of presentation and map geometry objects (see Items 3.3.5 and 3.3.6) and therefore also the process of cartographic generalisation.

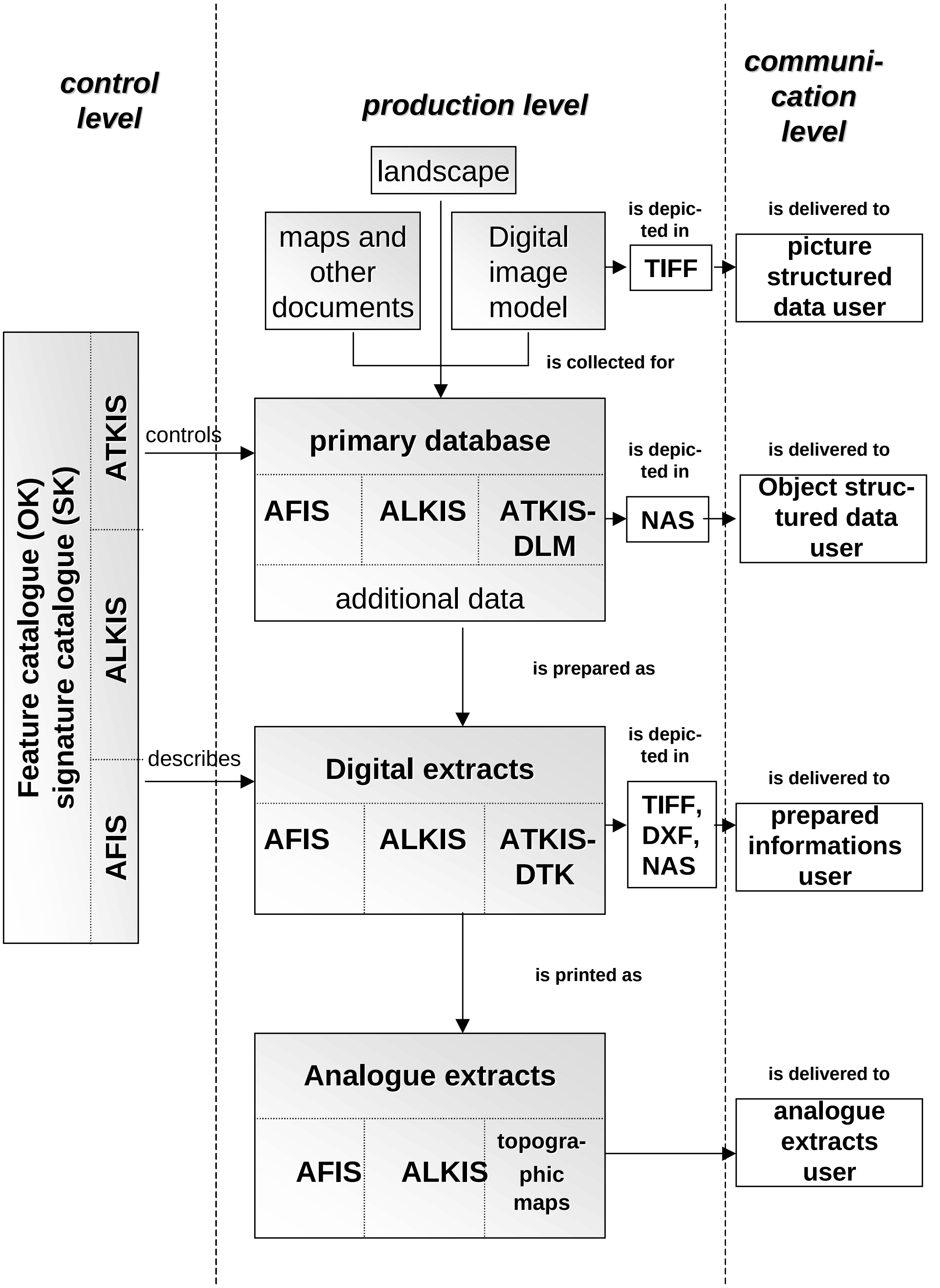


Figure 2-1: Common AFIS-ALKIS-ATKIS Reference Model

The primary database data are differentiated by the degree of abstraction by which they model the earth's surface and the associated situation. They show properties such as object structuring and geocoding. They contain both the **technical objects** with their semantic and geometric information and also the **additional data** required for presentation purposes:

- namely the **presentation objects** for text and signatures (see 3.3.5)
- and also the **map geometry objects** with the associated map geometry for a certain map scale that are linked to the topographical objects through a unilateral relation (see 3.3.6).

The data of the primary databases contain the complete description of technical objects, including the data on their cartographic or textual depiction in one or several target scales. The primary database data are thus modelled so that for presentation purposes, they can be depicted fully automatically, i.e. without further interactive intervention, in the intended output format.

At **communication level,** users are provided with object-structured or image-structured data, specially prepared information or analogue extracts that are able to hold the entire data content or extracts according to content and area and also management data for any number of time periods.

3 The Conceptual Model

3.1 Fundamentals of modelling

3.1.1 Norms and standards

International normalisation and/or standardisation activities in the field of geoinformation are currently being carried out in the following bodies:

- ISO/TC 211 Geographic Information/Geomatics
- Open GIS Consortium (OGC).

The goal is to create foundations for the common, holistic and cross-field use of geodata at various locations by individuals, applications and systems based on a standard description of the content of existing or planned data inventories, the functionalities of data processing and communication. The modelling is based on the results of ISO/TC 211 in the form of the 19100 series of standards at their current stage of processing. The data exchange interface (see Chapter 10) also uses parts of the OGC specifications.

3.1.2 Modelling and Description Language

The AdV decided to use the *Unified Modeling Language (UML)* for describing the application schema and the feature catalogue. This language is also used by ISO/TC 211 in the field of geoinformation standardisation.

UML was developed by the *Object Management Group* (OMG) for the purpose of describing application schemas. Semantics and notation of UML are described in the *UML Notation Guide*. In order to guarantee standard use of UML in the 19100 family of standards, their application is specified in ISO 19103 *Conceptual schema language.* The purpose is the complete and unambiguously interpretable, formal description of the content and structure of data inventories. The description is independent from type of implementation and the used programming language. A standard description of all geodata can be achieved with formal languages. The application schemas thus described can be automatically interpreted by suitable programs and translated into internal data and/or database structures.

A universal and system-independent data exchange and/or file format is automatically generated in conjunction with so-called encoding rules (Model Driven Approach). These encoding rules are created in accordance with ISO standard 19118 *Encoding* and the GML specification of the OpenGIS Consortium (OGC). The language XML (Extensible Markup

Language) of the World-Wide-Web-Consortium (W3C) is used as the format (see Chapter 10).

3.2 Task and Structure

The application schema provides the formal description for data structures and data content in one or several applications. It contains the complete description of a database and in addition to geographical data, may also contain other associated data. The fundamental concept of abstracting the real world means the introduction of thematic objects and of rules and regulations on how it is documented and managed. Thematic objects are classified by type. At the type level, the application schema describes the feature types of the real world. Data themselves exist at instance level. They represent individual examples of a feature type in the real world and can be interpreted by the application schema, see also ISO19101 *Reference model* and 19109 *Rules for application schema.*

The purpose of an application schema is to achieve a common and unified understanding of data and document the data content for a specific application environment so as to obtain unique information about these data.

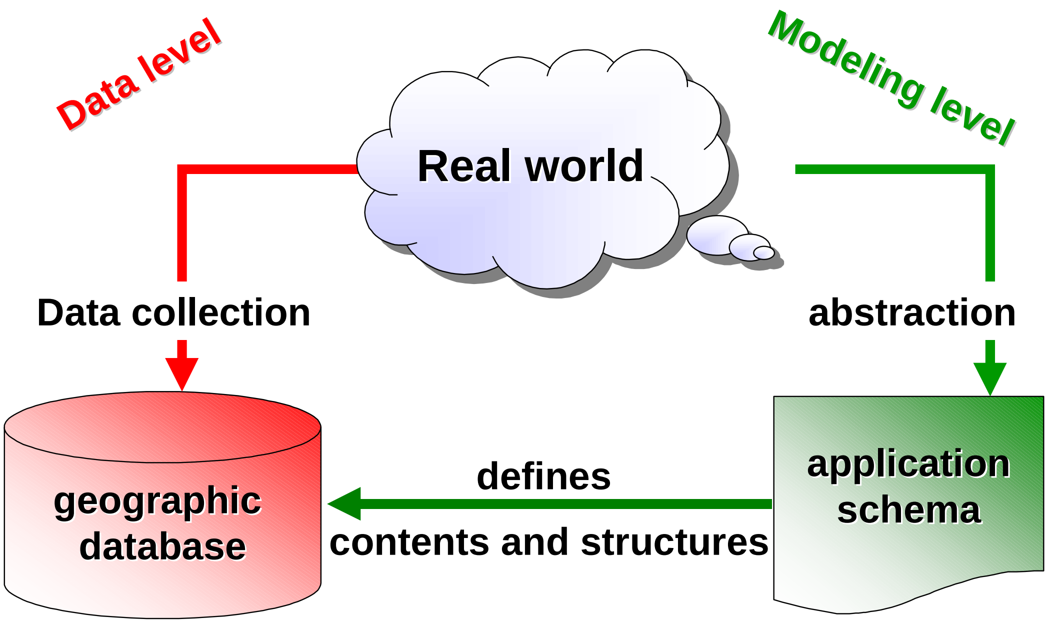


Figure 3-1: The role of the application schema

The common AFIS-ALKIS-ATKIS application schema offers a unified and object-oriented basic model for AFIS, ALKIS and ATKIS, which wherever possible is to be depicted and managed using the commercially available GIS software.

An application schema can use specifications from various sub-schemas. In the case of the AFIS-ALKIS-ATKIS application schema, mainly sub-schemas from the ISO 19100 series of standards are used. In those areas, where there are no ISO-standards up to now, additional schemata of the OpenGIS-Consortium are used.

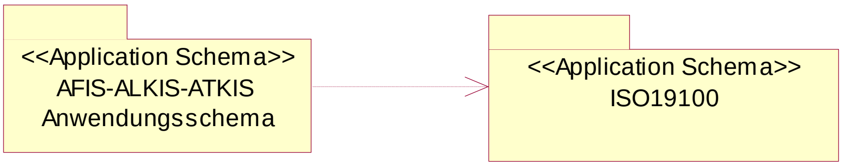


Figure 3-2: Dependency of the AFIS-ALKIS-ATKIS application schema on the structures standardised from ISO 19100

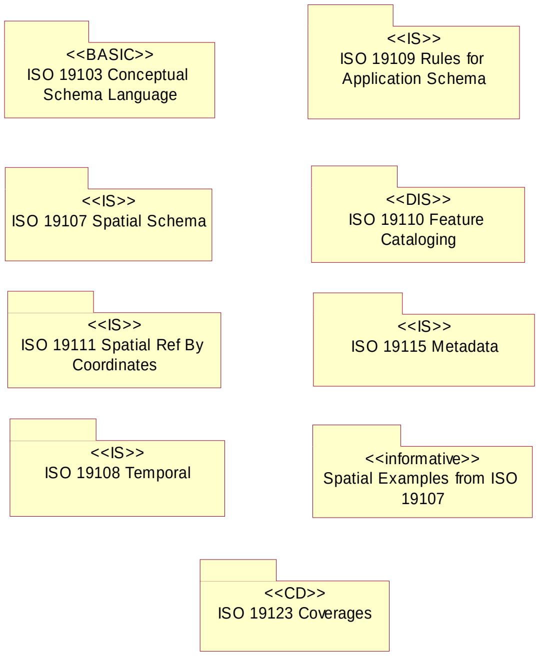


Figure 3-3: Used components from the ISO 19100 series of standards

The AFIS-ALKIS-ATKIS application schema is sub-divided into the basic schema (Section 3.3), the versioning schema (Section 3.4) and the AFIS-ALKIS-ATKIS thematic schema (Chapters 5 to 8). The basic schema is the basis on which thematic objects are modelled in the thematic schemas. The versioning schema shows the concept for historicising thematic objects. An internal schema is not part of common modelling. It is created by depicting a conceptual application schema in specific GIS systems as part of the implementation process.

The application schema is the basis on which operations for data exchange and technical stipulations for data outputs are defined.

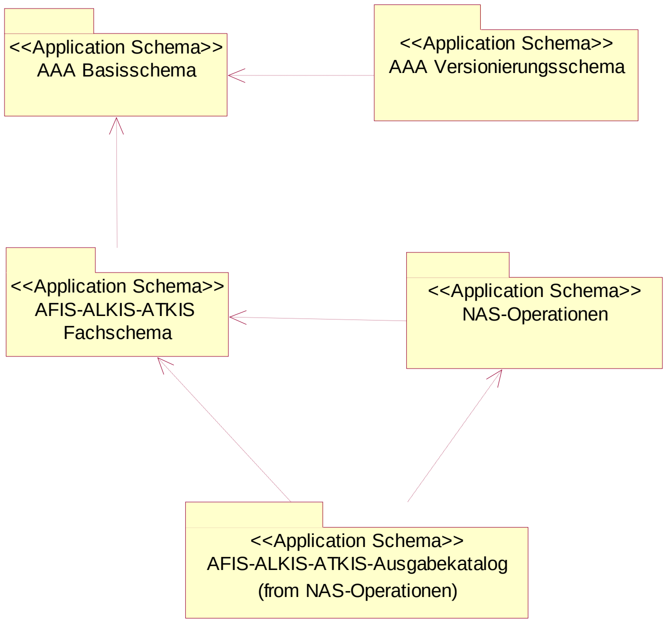


Figure 3-4: The components of the AFIS-ALKIS-ATKIS application schema

3.3 The AFIS-ALKIS-ATKIS Basic Schema

The AFIS-ALKIS-ATKIS basic schema (AAA basic schema) forms the basis for the technical modelling of AFIS, ALKIS and ATKIS objects and for the data exchange process. The thematic schemas are created from this basis. Its application is not limited to just AFIS, ALKIS and ATKIS. Other technical information systems can also use the classes defined in the basis schema for modelling their thematic schema.

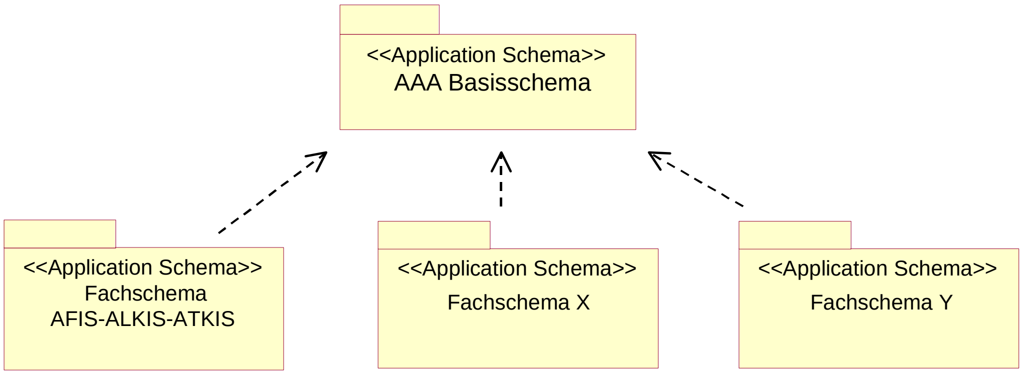


Figure 3-5: The basic schema as a basis for the modelling of application-specific thematic schemas (e.g. AFIS, ALKIS and ATKIS)

The basic schema is sub-divided into eleven packages - "AAA_Basisklassen" (BasicClasses), "AAA_Katalog" (Catalogue), "AAA_SpatialSchema", "AAA_GemeinsameGeometrie" (CommonGeometry), "AAA_UnabhaengigeGeometrie" (IndependentGeometry), "AAA_ExternalCodeLists","AAA_Praesentationsobjekte" (PresentationObjects), "AAA_Punktmengenobjekte" (PointCoverages), "AAA_Projektsteuerung" (ProjectManagement), "AAA_Nutzerprofile" (UserProfiles) and "AAA_Operationen" (Operations).

The packages AAA_Nutzerprofile and AAA_Operations only serve as a basis for a user management and a modeling of operations in the basic schema respectively. They only comprise empty, abstract classes, which have be concretizised in the respective thematic application schemas. Out of that reason, a further description of these packages is omitted.

The following systematics is used for unique designation of the defined classes:

1. Standardised classes maintain the standardised prefix in the class name (e.g. FC for "Feature Catalogue", MD for "Metadata")
2. Classes as AFIS-ALKIS-ATKIS-specific additions to the standardised *Feature Catalogue* geted the prefix AC
3. Classes with fundamental meaning for AFIS, ALKIS and ATKIS get the prefix AA
4. Classes derived from the ISO TS_*Component classes ("simple topology"), get the prefix TA; also the analogously created class for topological surfaces with multiple spatially separated geometries (TA_MultiSurfaceComponent)
5. Classes with commonly used geometries get the prefix AG
6. Classes of independent geometries get the prefix AU
7. Classes of presentation objects get the prefix AP
8. Classes for the modelling of PointCoverages get the prefix AD

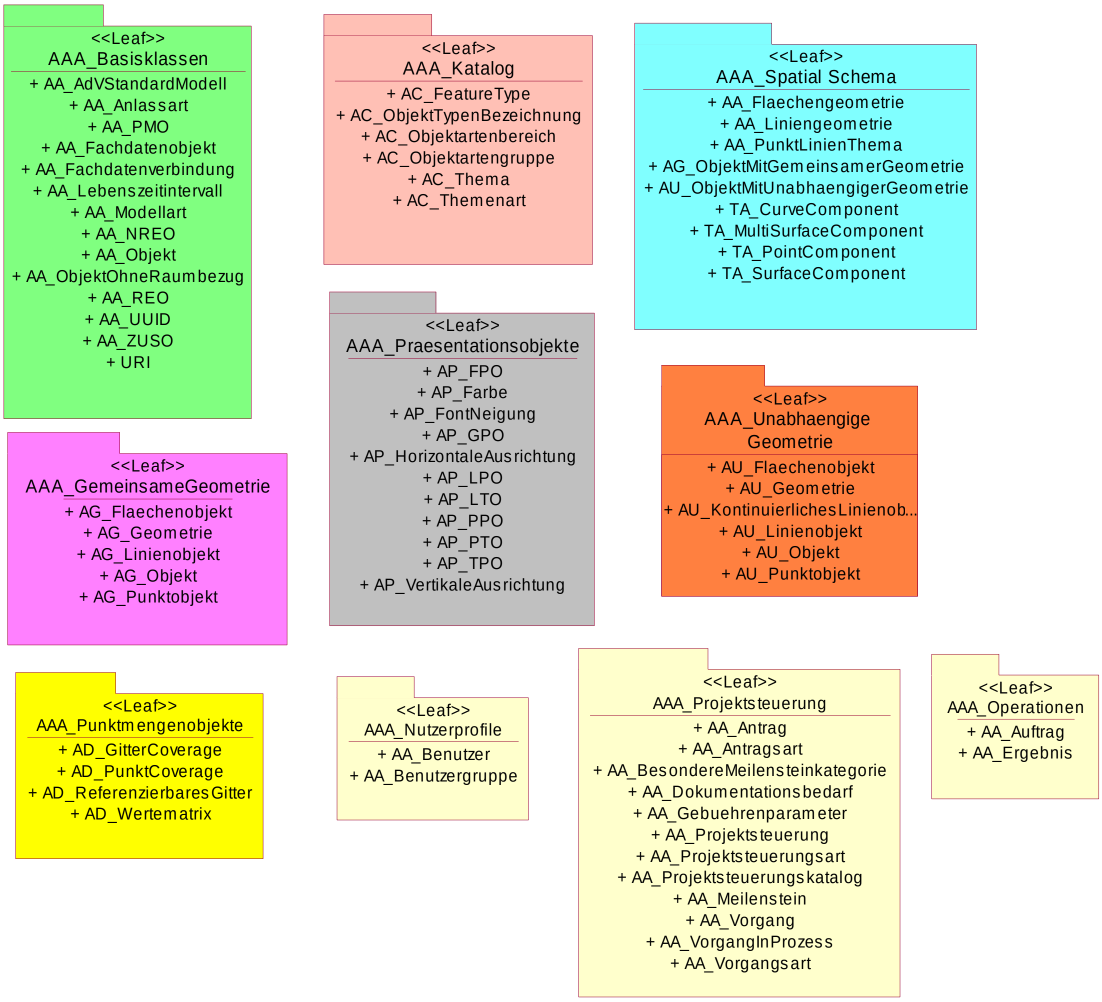


Figure 3-6: Components of the basic schema

3.3.1 Fundamental Principles of Object Formation

The rules for creating application schemas are defined in standard 19109 *"Rules for Application Schema"* of ISO/TC 211. This standard also comprises the general model for the description and formation of thematic objects (*General Feature Model).* The common basic schema is connected to the *General Feature Model* of ISO 19100 and this is expanded by the meta class "AA_ObjectWithoutSpatialReference", in order to be able to form object classes for which no spatial reference is permitted.

Independent objects are formed on the basis of the thematic object view. **Objects with geometrical characteristic** can bear point, line or surface geometries or be of type PointCoverage. **Objects without spatial reference** (e.g. persons) bear no geometry and cannot be determined at a specific location. They can, however, be associated with other spatially-referenced and non-spatially referenced objects, e.g. land parcels, buildings or addresses.

For systematisation and to support the creation of the thematic schemas, 4 general types of object characteristics are pre-defined in the common AAA basic schema:

- Spatially-referenced elementary objects (AA_REO)
  Spatially-referenced elementary objects are to be formed if, in addition to technical properties, geometric or topological properties shall also be shown.
- Non-spatially referenced elementary objects (AA_NREO)
  Non-spatially referenced elementary objects are to be formed if only technical but no geometric or topological properties are to be shown.
- Composed objects (AA_ZUSO)
  Composed objects are formed in order to create the correlation between any number and combination of semantically associated, spatially-referenced elementary objects, non-spatially referenced elementary objects or composed objects. However, a composed object must have at least one object as a component.
- PointCoverages (AA_PMO)
  For certain kinds of thematic objects, which consist of a large number of geometric locations with the same attributes for each location (e.g. Digital Terrain Models, temperature and pressure distribution), it is more adequate to use coverage objects instead of single REOs for each point. A Coverage Object is a function which maps geometries to their attribute values.

Elementary objects are the smallest possible thematic unit. The formation of object components or lines as object components with thematic information like for the previous systems ALK and ATKIS has been abolished.

Management of the **History of Objects** (see Chapter 3.4) is supported. **Integration and interconnection between the thematic objects and technical data** of other technical areas are also supported (see Chapter 3.3.7).

All thematic object classes that can be instanced are to be derived in the application-related sub-schemas from the following object classes of the basic schema (see 3.3.4.1 to 3.3.4.4) by inheritance.

- AA_ZUSO
- AA_NREO
- TA_PointComponent
- TA_CurveComponent

- TA_SurfaceComponent
- TA_MultiSurfaceComponent
- AG_Objekt
- AG_Punktobjekt (PointObject)
- AG_Linienobjekt (LineObject)
- AG_Flaechenobjekt (SurfaceObject)
- AU_Objekt
- AU_Punktobjekt (PointObject)
- AU_Linienobjekt (LineObject)
- AU_KontinuierlichesLinienobjekt (ContinuousLineObject)
- AU_FlaechenObjekt (SurfaceObject)
- AD_PunktCoverage (PointCoverage)
- AD_GitterCoverage (RectifiedGridCoverage)

For presentation objects (see Item 3.3.5) the following object classes of the basic schema can be directly used and/or instanced.

- AP_PPO
- AP_PTO
- AP_LTO
- AP_LPO
- AP_FPO

Other object classes that can be instanced are also permitted to be derived from this object class of the basic schema through inheritance.

3.3.2 Attributes

The objects to be described in the thematic schemas can have self-referenced attributes. Attributes bear the statistical information of the objects. Attributes are always defined by a name and a value type. Value types can be either basisdata types (numbers, character strings, date and time data) and also complex data types such as geometries or quality features. **Attributes** can basically be multiple and character strings can have any length.

Attributes of type Date, Time or DateTime are modelled according to ISO 8601, chapter 5.4.1 and chapter 5.3.3. Accuracy of the time is the full second, time zone is always UTC (Universal Time Coordinated, also known as Greenwich Mean Time, abbreviation: Z). Example: 2004-04-01T17:06:31Z

3.3.3 Associations

The objects to be described in the thematic schemas can have externally-related attributes (associations and/or relations). Various types of associations can be used in the thematic schemas.

- According to the ISO *General Feature Model*, thematic objects can enter into any number of associations. These are defined in the thematic sub-schemas.
- In addition, several associations between objects are already specified in the common basic schema.
- Relation for forming composed objects (ZUSO)
  A ZUSO is composed of at least one object. The brackets around these objects form the association *bestehtAus (isComposedOf)* between "AA_ZUSO" and "AA_Object".
- Underpass relation
  Underpass relations *hatDirektUnten (hasBelow)* are used in order to show a relative vertical position of individual objects in relation to other objects. It is not possible to indicate an absolute vertical position using underpass relations, because such relations always contain only the relation between the participating objects.
- Map geometry
  The relation between map geometry objects (=generalised geometry, see Item 3.3.6) and the associated basic objects *istAbgeleitetAus* (*is_derived_from*) indicates the objects from which the map geometry objects are derived.
- Thematic data connection
  If an AFIS, ALKIS or ATKIS object shall point at a technical data object being managed in an external technical data system, this can alternatively be described through the *verweistAufExternes(indicatesExternal)* attribute.
- Depiction relation
  Presentation objects serve to depict objects of the primary database data. This correlation is shown by the *dientZurDarstellungVon (isUsedtoDepict)* reference between the presentation object and other objects.

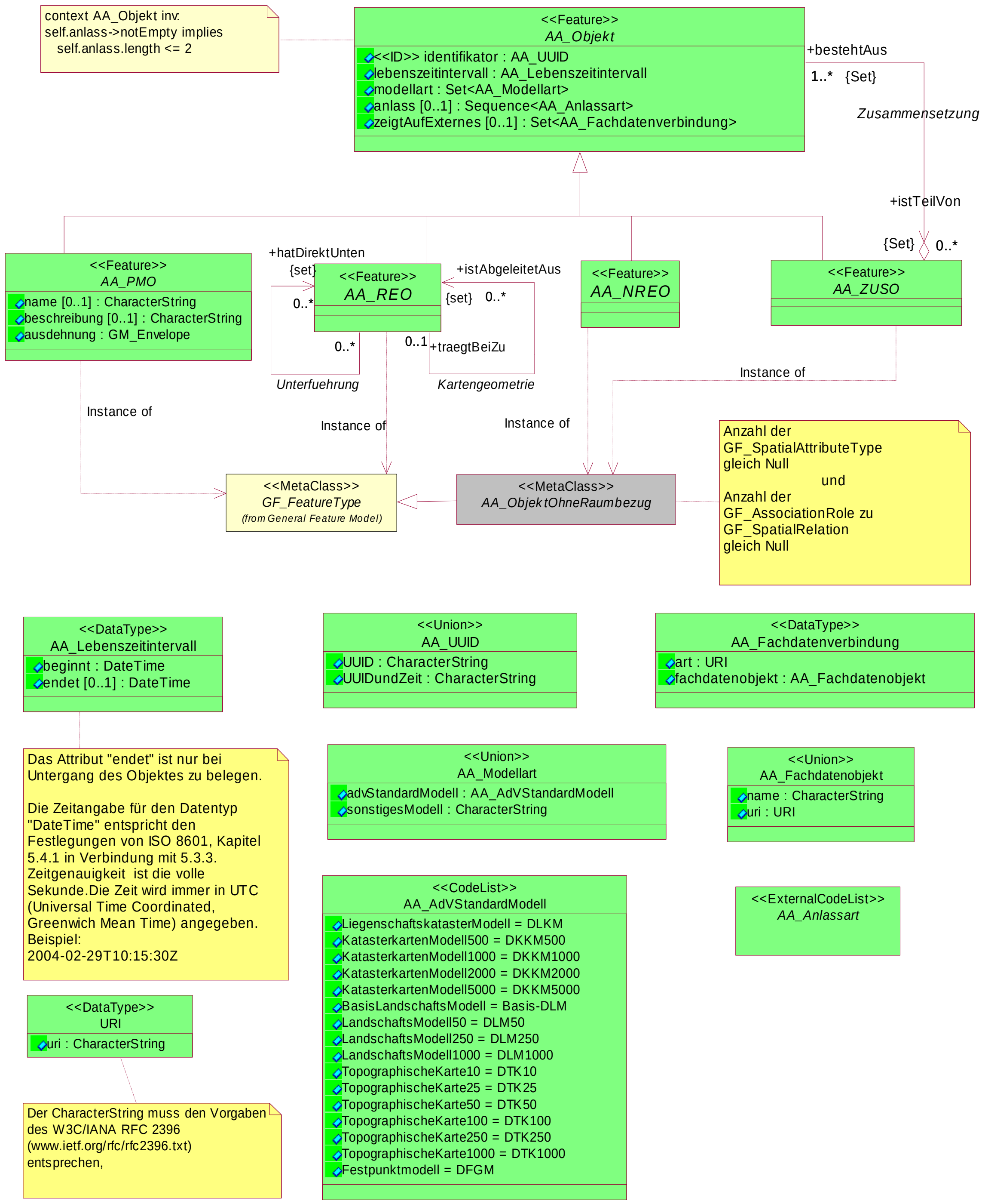


Figure 3-7: Modelling of basic classes

3.3.4 Spatial reference, geometry

3.3.4.1 Fundamental principles

ISO standard 19107 *Spatial schema* provides spatially-referenced basic constructs for use in application schemas; for simplification purposes, exclusively the following constructs are used for AFIS, ALKIS and ATKIS:

Geometric objects (GM_Object)			Topological objects (TP_Object)	
Geometric Primitives	**Geometric Complexes**	**Geometric Aggregates**	**Topological Primitives**	**Topological Complexes**
GM_Point GM_Curve GM_PolyhedralSurface	GM_CompositeCurve GM_CompositeSurface	GM_MultiCurve GM_MultiSurface	TS_PointComponent TS_CurveComponent TS_SurfaceComponent TS_Face	TP_Complex

The geometric and topological objects are described as UML classes. The standard also contains spatial operations that use geometric and topological objects *(GM_Object and/or TP_Object)* as parameters (create, delete, change, spatial evaluations...). The defined classes have no direct application, i.e. they cannot be instantiated. Their use in special application schemas is achieved through inheritance; insofar as the classes of the *Spatial Schema* for AFIS, ALKIS and ATKIS are not supplemented by special attributes, they are however directly used in this application for simplification purposes.

The geometric primiteves usually appear as attributes of objects; this does not mean, however, that the geometry is always redundant in principle. The common AFIS-ALKIS-ATKIS application schema has the following options for linking the spatial reference:

- The formation of node-shaped, edge-shaped and face-shaped objects with "simple topology". In addition face-shaped objects with "simple topology", which consist of two or several spatially separated faces (required for modelling of land parcels with multiple, non adjacent parts).
  The ISO schema "Simple Topology" is used, which expresses topological features by geometrical features, while still offering topological functionality. (see 3.3.4.2)
- Formation of point, line and surface objects, which share lines and surfaces. (see 3.3.4.3)
- Formation of point, line and surface objects with "independent" geometry. (see 3.3.4.4)

- Formation of topological and geometrical "topics" that allow to selectively combine feature types to so-called complexes, in order to express geometric identities and/or topological correlations.

Each spatially-related AFIS-ALKIS-ATKIS thematic object (AA_REO) refers to a maximum of one geometry. Should it become necessary to hold several geometries to form a real-world object (e.g. generalisation, various co-ordinate reference systems, point and surface geometry), a separate thematic object (where necessary as a map geometry object) shall be formed in each case.

The necessary expansions and limitations of the ISO *Spatial Schema* are summarised in the following diagrams.

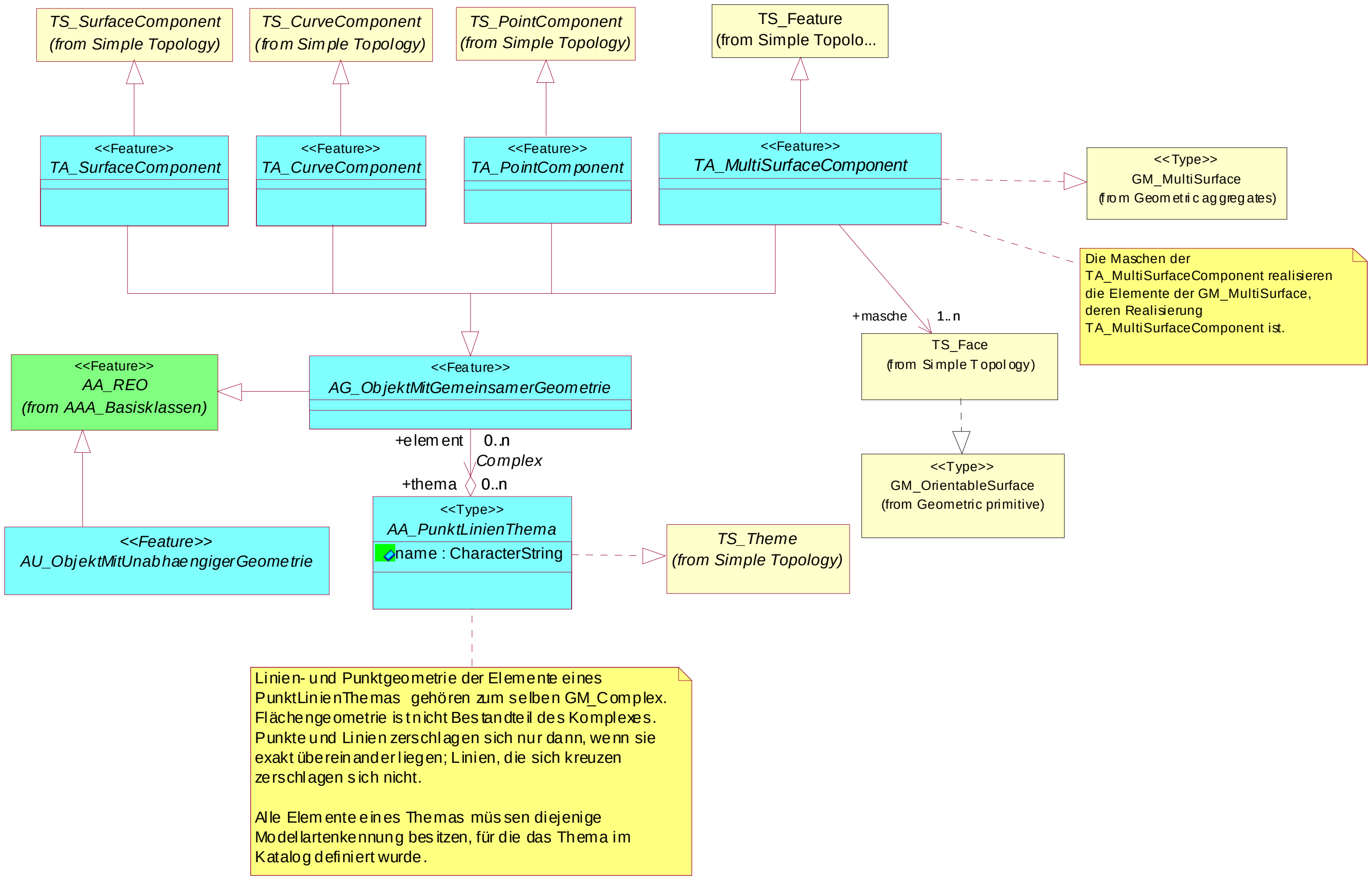


Abbildung 3-8: Summarizing representation of the required supplements for AFIS-ALKIS-ATKIS to the standardised Spatial Schema

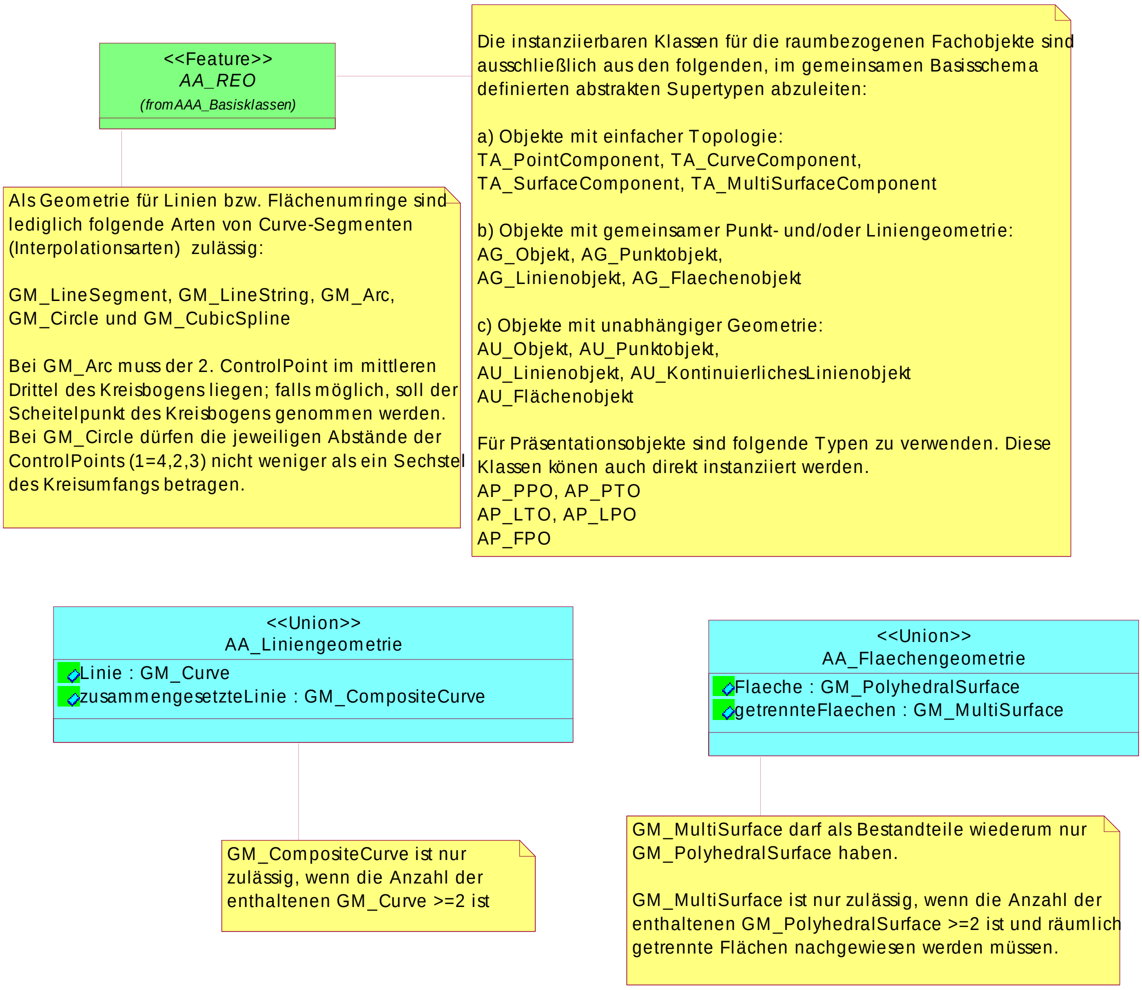


Figure 3-9: Restrictions in respect of geometry and instantiable classes

3.3.4.2 Features with simple topology

ISO 19107 *Spatial Schema* offers the *Simple Topology* schema as a module for an application schema. On this basis, features are provided that express topological properties by geometrical properties. The basic schema provides the TA_*Component as an application of this module. In addition to the corresponding classes of the *Spatial Schema*, these classes also offer common AFIS-ALKIS-ATKIS object properties (identifier, lifetime interval, cause of change) and also the option of linking various feature types via the construct of the "PointLineTheme" by geometry. The TA-classes may belong to a topological theme and one or several PointLineThemes at the same time. The TA_MultiSurfaceComponent class has been defined differently to the TA_SurfaceComponent class to enable the referenced rings *(TS_Face)* to realise spatially separated surfaces *(GM_OrientableSurface)*. This also permits the topological modelling of exclaves. Exclaves should therefore not be modelled by feature to feature relations (Relation *Composite* [*composite* > *component*] between TS_Feature and TS_Feature).

3.3.4.3 Objects with commonly used geometry

The "AAA_CommonGeometry" package provides the basic classes for features, whose geometries consists of points, lines and surfaces that share their geometries. This involves using the properties of the expanded "AAA-SpatialSchema", which also provides the "PointLineTheme". Furthermore, in accordance with ISO 19107 and 19109 the geometry is expressed by geometrical and topological primitives (GM_PointRef and GM_CompositeCurve) provided for the common use of geometry . Thus, the geometry-carrying primitives (GM_Point and GM_Curve) are connected relationally to the features and can therefore be commonly used by several features. The common use of geometry relates only to points and lines, not to surface geometries. Lines are joined and grouped for a common use if all their vertices are identical and have equivalent interpolations; lines that cross do not split. Basic classes "AG_Objekt", "AG_PunktObjekt", "AG_LineObject" and "AG_FlaechenObjekt" are to be used for defining spatially-related feature types with common geometry.

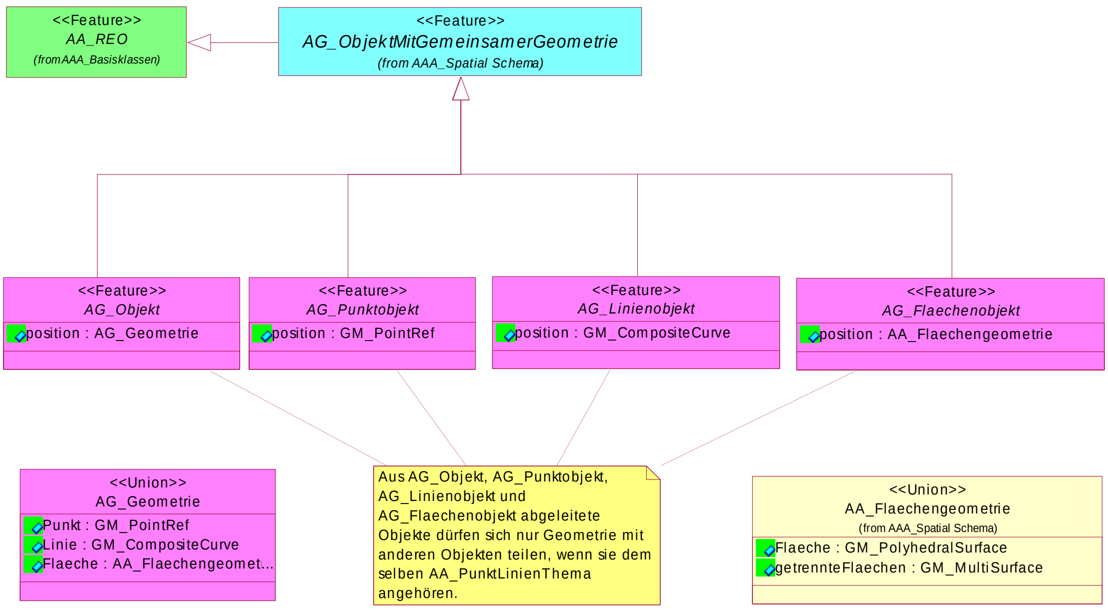


Figure 3-10: Features with common geometry

3.3.4.4 Features with independent geometry

The "AAA_IndependentGeometry" package provides five basic classes of features, whose geometry consist of independent points, lines and surfaces. These basic classes are to be used as a basis for spatially-related feature types with independent geometry (e.g. presentation objects).

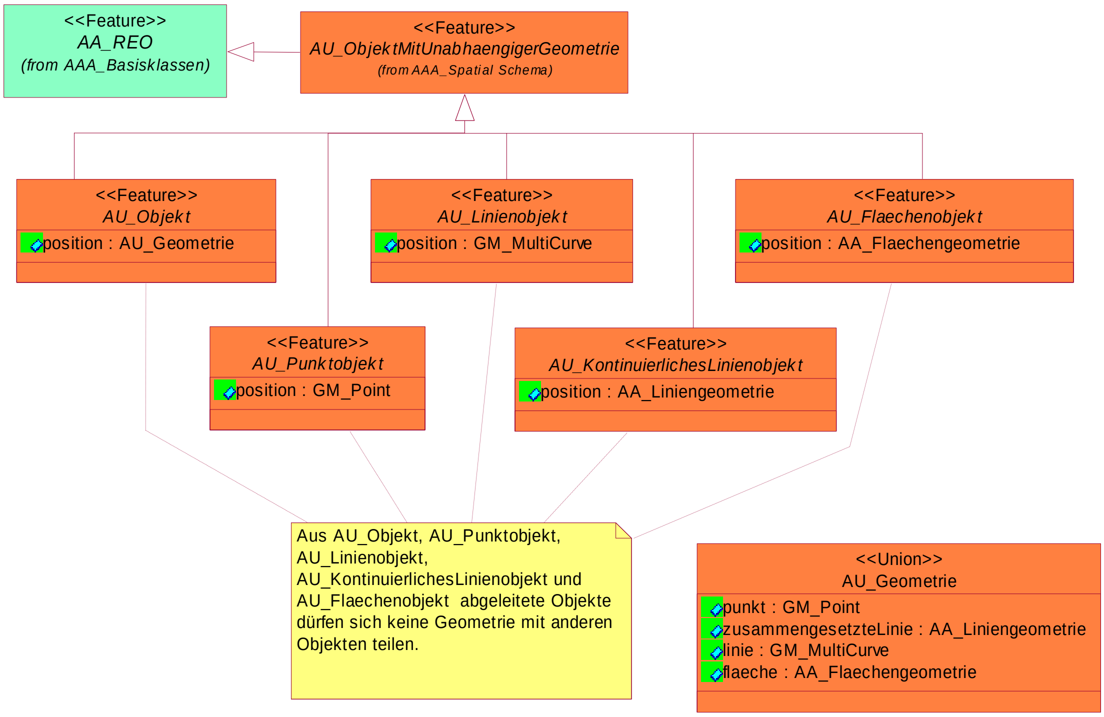


Figure 3-11: Features with independent geometry

3.3.4.5 Spatial reference system, coordinates

For each geometry the particular coordinate reference system (CRS) can be indicated in AFIS-ALKIS-ATKIS.

De-jure standard ISO 19111 *(Spatial Referencing by Coordinates)* determines the type of definition and description of coordinate reference systems. There are several types of reference systems, depending on whether a reference to an ellipsoid exists, whether longitude and latitude data is used or whether reference is made to projected coordinate systems. For additional management of **elevation data and/or coordinates** without reference to an ellipsoid (e.g. mean sea level elevation), a so-called compound coordinate reference system (CCRS) can be used. This will then consist of the description of the reference system for the position and the description of the reference system for the elevation. The coordinate reference systems commonly used in Germany are listed in the document “**Koordinatenreferenzsysteme und Maßeinheiten für AFIS-ALKIS-ATKIS**” (Coordinate reference systems and Units of Measure for AFIS-ALKIS-ATKIS) with their designations and abbreviations.

The type of coordinate reference system determines the number of **coordinate values** contained in the points of the individual geometries (e.g. easting, northing or easting, northing, elevation). The use of compound reference systems may be limited in feature classes.

Because the current ALK and ATKIS systems do not contain details on the CRS the data depends on, a task of the migration concept is to define the relevant specifications.

3.3.5 Portrayal, presentation objects

The complete description of features comprises the following components

- Semantics (factual data, attributes, values),
- Spatial reference (geometry, topology) and
- Presentation (text, signature).

The presentation depends on the correlation between information and models. This correlation can be stored multiply feature-related in the *model-typen* attribute. It should in principle be assumed that the presentation can be obtained automatically by evaluating the primary database data. If the required functions are so complex that they cannot be economically realised or result in uneconomic processing times, additional information must be held in the database for the presentation. These information are referred to as **presentation objects.** Through the model-type, these can be differentiated from normal AFIS-ALKIS-ATKIS primary database data and can accordingly be evaluated.

As **presentation objects,** all text and map portrayals are defined, which cannot be fully-automatically generated and positioned for a specific target scale. The signature numbers used in the presentation objects for controlling the presentation are defined in portrayal catalogues.

The common basic schema provides the presentation objects defined in the following illustration.

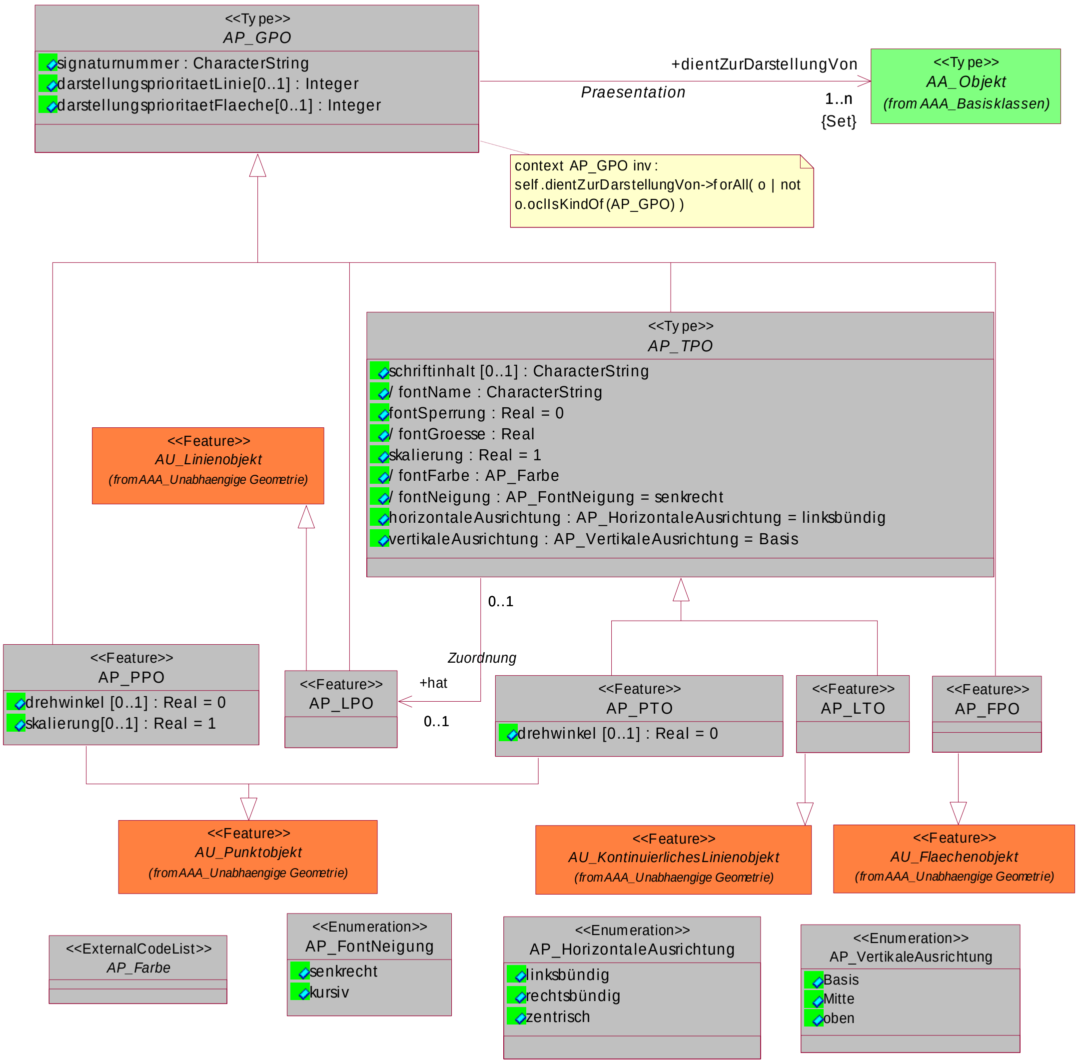


Figure 3-12: Presentation objects

The *AAA_Presentation objects* package uses the features of *AAA_Independent geometry* for the purposes of presentation.

The portrayal regulations are to be defined in portrayal catalogues. The **portrayal catalogues** can be structured formalised in accordance with the regulations of de-jure standard ISO 19117 *Portrayal*.

3.3.6 Map geometry objects

As **Map geometry objects** are those features defined, whose geometric form and/or position have changed on the derivation for a specific map scale for reasons of cartographic generalisation. A map geometry object has to contain at least the following independent information: The identifier, the map model data, e.g. DTK10, to which it belongs, the unilateral relation *is_derived_from* to the fundamental AFIS-ALKIS-ATKIS object and the geometry itself. It may also contain the attributes of the fundamental AFIS-ALKIS-ATKIS object, in order to be evaluated for the presentation.

3.3.7 Object for Point Coverages

A thematic object class is defined as a coverage object if it comprises a lot of geometric locations with identical attributes for each location. Within the context of AAA this is mainly the case for Digital Terrain Models, which comprise elevation values for defined points in a recitified grid. Additionally the possibility is needed to store measured elevation values for arbitrary point locations. Out of that reason, beside a class for gridded coverages (AD_GitterCoverage) a class for arbitrarily distributed points (AD_PunktCoverage) is modeled. The modeled classes realize the respective classes of ISO 19123 Coverages with the following restriction: The sequence of the attribute values within the grid (CV_SequenceType) must be “linear”.

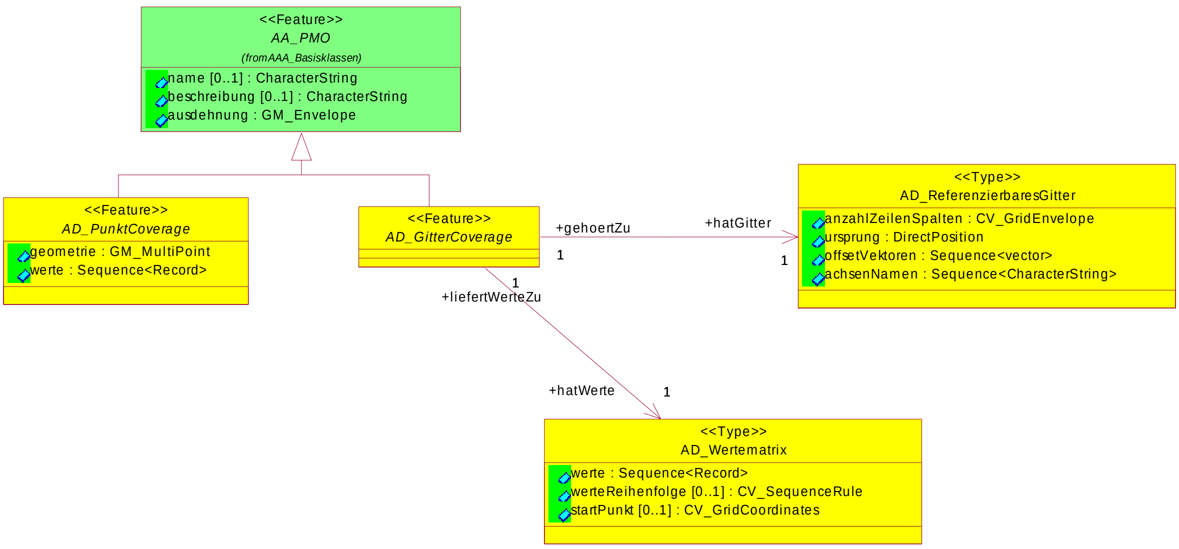


Figure 3-13 Modelling of the Coverage Classes

3.3.8 External code lists

Code lists used in the basic schema, which due to their character a) have to be filled by the application-specific subschema and b) have to be expandable for the integration of various applications, are defined as abstract classes the basic schema. They have to be concretizised in the thematic schemas Expansions and changes to these lists do not result in a new version of the data exchange specification (see Chapter 10). They are held and maintained in the form of so-called *dictionaries* at a central location with the option of online access.

3.3.9 Identifiers, links

Identifiers act on behalf of the object that they represent. The essential properties of an identifier are:

- It is "system-wide" unique, whereby the associated definition of "system-wide" is able to satisfy the requirements for state-wide and theme-wide uniqueness.
- Its appearance indicates that an object has been generated.
- It remains unchanged during the lifetime of an object.
- Its obsolescence indicates that an object no longer exists.

Thus the life-cycle of identifiers is identical to the life-cycle of the objects they represent. The question as to when identifiers may and may not be changed cannot therefore be answered from an IT point of view; factual criteria must be stated:

- when an object is created,
- which changes it can withstand without losing its identity and
- when it becomes obsolete.

For all objects, a unique designation is used as an object identifier. The identifier is structured as follows:

	Parts	**Meaning**	**Definition**
1	Global, unique code (2 characters)	Nationality	"DE" for Germany
2	Prefix (6 characters)	Code for the implementation or database generating the identifier and also for preliminary identifiers.	The characters start from the left with the abbreviations of the German states standardised in de-jure standard ISO 3166-2"Country Subdivision Code" (ISO, 15th December 1998). For Federal Agencies, the abbreviation "BU" is used and/or in the case of the Federal Agency for Cartography and Geodesy "BKG"; further digits are specified by the corresponding state and/or the federal agencies or the BKG. If during the processing for use of complete identifiers, preliminary identifiers are required, these start from the left with "*". This gives rise to the following table. Baden-Württemberg "BW" Bavaria "BY" Berlin "BE" Brandenburg "BB" Bremen "HB" Hamburg "HH" Hesse "HE" Mecklenburg-Western Pomerania "MV" Lower Saxony "NI" North Rhine-Westfalia "NW" Rheinland-Palatinate "RP" Saxony "SN" Saxony-Anhalt "ST" Saarland "SL" Schleswig-Holstein "SH" Thuringa "TH" Federal agencies "BU" Federal agency for cartography and geodesy "BKG" Preliminary identifier "_" Permitted characters are: A-Z, a-z, 0-9, _ without umlaut and ß
3	Suffix (8 characters)	Continuous numbers	Permitted characters are: A-Z, a-z, 0-9 without umlaut and ß

Examples of identifiers are:
"`DENW123412345678`" (definitive identifier)
"`DE_000000000001`" (preliminary identifier)

To realise a geodata infrastructure as defined by the OpenGIS Consortium (OGC) and using its interface definitions, all involved states and agencies have to define a system for the assignment of identifiers and a service interface, in order to ensure that objects can be found via their identifiers without requiring further knowledge. A common service is offered within the meaning of a cross-state solution; the system of assignment and distribution can still be defined state-specific.

In order to build up relations between objects in data exchange, identifiers are also used as references to objects.

Identifiers are also required to state which objects should be deleted and which objects should be overwritten when formulating updates. Because in these cases the objects have to be addressed in their concrete version, the aforementioned identifier will be supplemented by the creation date/time of the addressed object version. (See 10.1.3).

An important precondition for the common management of databases of various origins is that the integration situation is shown in the form of references between the data of the state survey and the technical data (**link).** This linking can take place unilaterally in the spatially-related basic information systems of the surveying authority or in the technical information system (unilateral linking) or reciprocally in both information systems (reciprocal linking). As linking properties, unique designations are to be designed and managed. These may also consist of the aforementioned identifiers and / or technical codes of the respective databases.

3.4 History, version concept

With the AFIS-ALKIS-ATKIS data, it is sometimes necessary to manage versions and historical data. The scope of use depends on the information system and its application in the states. One essential application for the version concept is the procedure for a user-related update of the primary database (NBA).

The version concept has been defined in consideration of the following modelling principles:

- No distinction is made in the application schema between current and historical data, i.e. no separate historical feature types are formed for the full history.
- The historical as well as the current information (version) is stored for each object.
- The partially redundant storage of object attributes in several versions is accepted in return for faster data access to the corresponding version.

The version concept assumes that each thematic object carries an identifier, attribute and relation, as well as a lifetime interval (creation and expiry date). Entering an object into the primary database data generates the first version of the object and registers it in a feature version container for feature versions. If a non-object forming property changes due to a updating, a new version of the object is generated. The historized version does however remain within the container for feature versions, i.e. the identifier does not change. The creation date of the new version is the same as the expiry date of the previous version. The individual versions of an object can be clearly distinguished using the lifetime interval. By evaluating the various versions of an object, all changes can be determined in relation to any time period.

If object-forming properties change during a updating, this results from a technical point of view to the expiry of an object. The object is historized by assigning an expiry date to the last version. The object remains within the database. At any point in time, a version has all attributes and relations valid at that time. By “bracketing” the versions within a container for feature versions, the thematic object view remains in place.

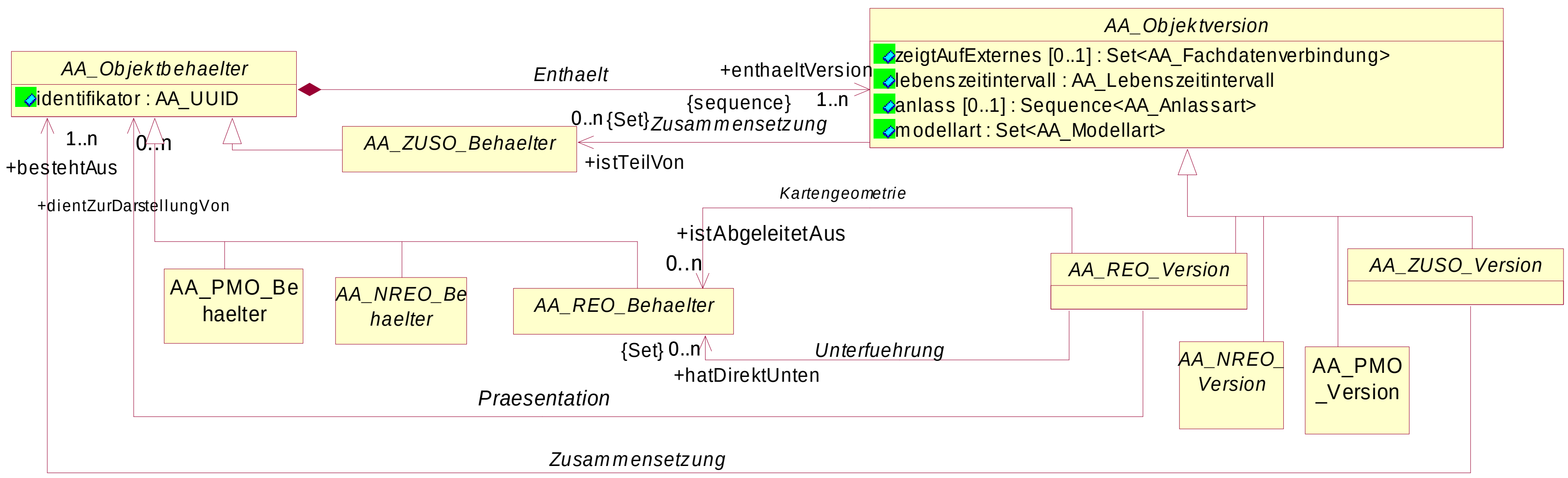


Figure 3-14: Versioning schema

Example of a version concept

Changes to attributes

Mrs Hilde Huber is registered in ALKIS at time t_1, i.e. a new object of the *Person* feature type is created:

	Identifier	Time interval		Surname	Christian name	has_address
		Start	End			
Version 1	DEBU5t44dFzb70Lg	t_1	t_∞	Huber	Hilde	DEBUf88FFgVc761s

Time ‘t_∞‘ means that the technical expiry of the object and/or version will be in the future. At time t_2, Mrs Huber changes her name to Meier, i.e. object “DEBU5t44dFzb70Lg“ of feature type *Person* creates a new version due to the change of the *Name* attribute:

	Identifier	Time interval		Name	Christian name	has_address
		Start	End			
Version 1	DEBU5t44dFzb70Lg	t_1	t_2	Huber	Hilde	DEBUf88FFgVc761s
Version 2	DEBU5t44dFzb70Lg	t_2	t_∞	Meier	Hilde	DEBUf88FFgVc761s

The time at which Version 1 expires is identical to the date on which Version 2 of the object is created. At time t_x, Mrs Meier sells her only plot of land. Because she has no other role within ALKIS, the object expires from a technical viewpoint

	Identifier	Time interval		Name	Christian name	has_address
		Start	End			
Version 1	DEBU5t44dFzb70Lg	t_1	t_2	Huber	Hilde	DEBUf88FFgVc761s
Version 2	DEBU5t44dFzb70Lg	t_2	t_3	Meier	Hilde	DEBUf88FFgVc761s

Version 2 and therefore the entire object is historized, not deleted.

Each new version of an object is assigned its own relations, on which it is based. Relations always start from a particular version of the object, i.e. a relation from one version to another object is valid only for this version. All cardinalities specified in the feature catalogue are retained in this way.

This is explained in Figure 3-14. Mrs Hilde Huber, of 17 Ottostraße 17, Munich is registered in ALKIS at t_1, i.e. an object of feature type *Person* and an object of feature type *Address* are created. At time t_2, Mrs Huber changes her name and from that point onwards is called Meier. A new version of the *Person* object will be created.

In the Figure, the arrows represent a relation. The direction of the arrow also indicates the direction of the relation. The new version of the *Person* object is in turn assigned a relation to the associated *Address* object. However, no new version of the *Address* object is created, because the relation to the *Person* object remains unchanged. A new version of the *Address* object would not cause a change to the *Person* object, e.g. when correcting in input error.

This example also shows that a relation always points from the version via the identifier to the container for feature versions and not to a version. The container for feature versions therefore forms a type of bracket around its various versions.

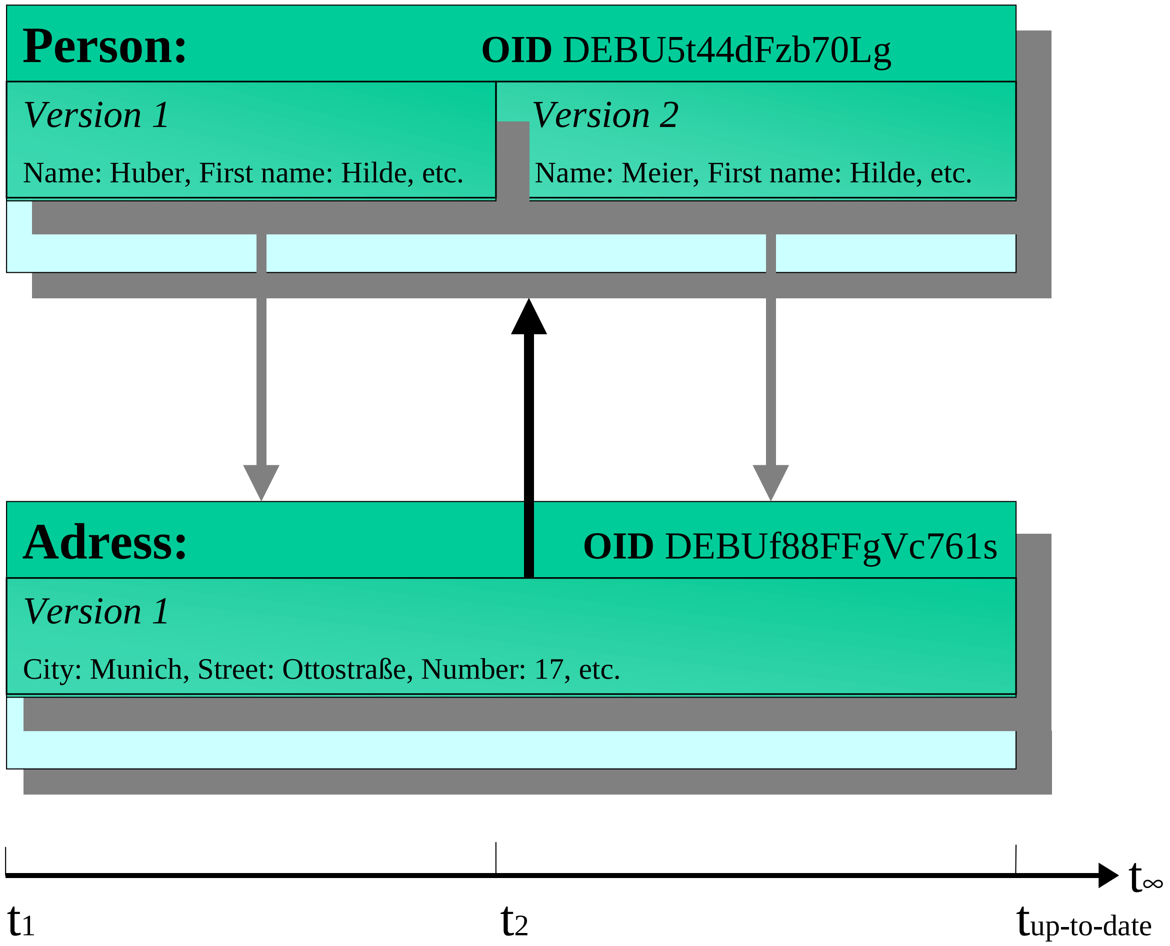


Figure 3-15: Example of versioning following attribute change

This technique can be used only to show relations that relate to the current version of the participating objects. If this is insufficient in a specific case, a version can exceptionally be directly referenced whereby the identifier in the reference should be supplemented by the time stamp for that version.

Changes to relations

Changes to relations result in versioning of objects as do attribute changes. Relations always change when the object to which the relation points is re-created, exchanged or removed.

This is explained in a modified example of Figure 3-14. At time t_3, Mrs Hilde Huber moves from 17 Ottostraße, Munich to Platanenallee 34a, Berlin. The *Address* object is exchanged with OID "DEBUf88FFgVc761s", to which the *has_address* relation points from the *Person* object (new OID "DEBUk41233THjbkO"). Thus, the relation associated with the *Person* object changes and the *Person* object must be versioned.

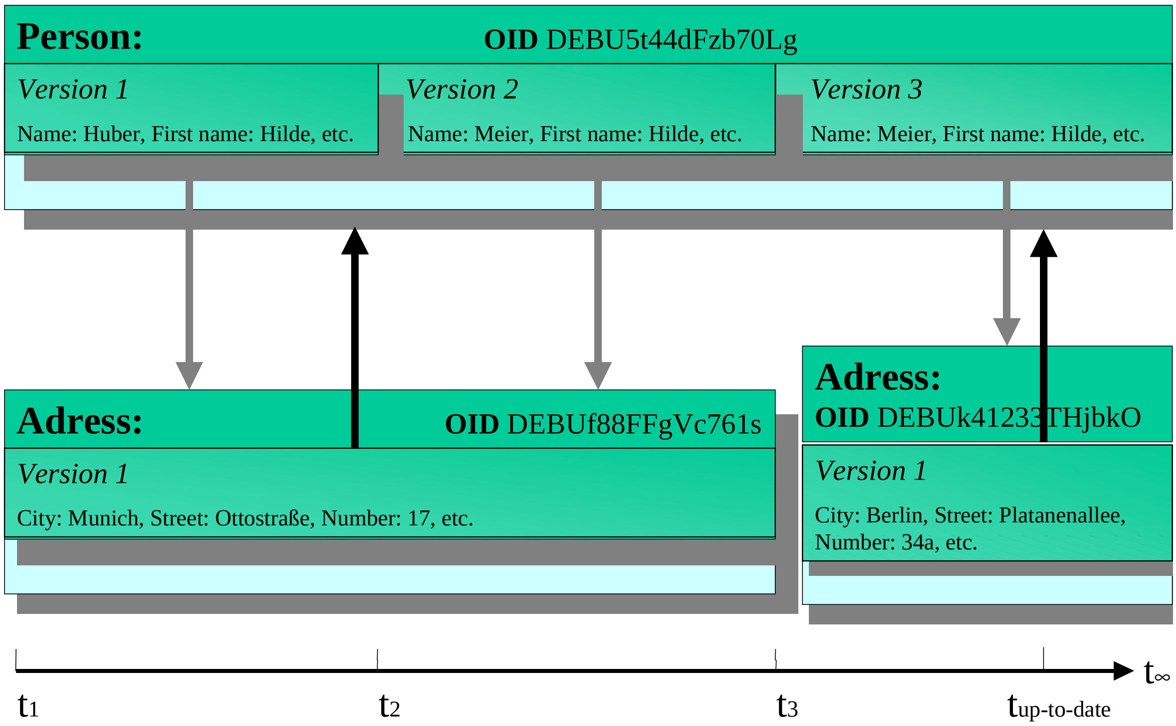


Figure 3-16: Example of new version following relation change

The following table shows the pattern:

	Identifier	Time interval		Name	Christian name	has_address
		Start	End			
Version 1	DEBU5t44dFzb70Lg	t_1	t_2	Huber	Hilde	DEBUf88FFgVc761s
Version 2	DEBU5t44dFzb70Lg	t_2	t_3	Meier	Hilde	DEBUf88FFgVc761s
Version 3	DEBU5t44dFzb70Lg	t_3	t_∞	Meier	Hilde	DEBUk41233THjbkO

3.5 Quality and metadata

The common AFIS-ALKIS-ATKIS data model records and manages quality and metadata on the basis of the following ISO standards

- ISO 19109 Geographic Information – Rules for Application Schema
- ISO 19113 Geographic Information – Quality Principles,

- ISO 19114 Geographic Information – Quality Evaluation Procedures and
- ISO 19115 Geographic Information – Metadata

The **quality data** is differentiated by non-quantifiable summary information (purpose, application and history) and quantifiable information (data quality elements *completeness, logical consistenc as well as, geometric, content and temporal accuracy).*

The quality information is detailed as metadata in accordance with ISO 19115 and moreover for quantitative, aggregated quality data if required in the form of detailed quality evaluation protocols in accordance with ISO 19114.

Metadata are "data on data" and describe geodata in terms of user-relevant aspects for evaluating the suitability of the data and access to the same. ISO distinguishes between 40 optional, mandatory and conditionally mandatory metadata elements, sub-divided into entities and also into the following sections:

- Identification,
- Data quality,
- Updating,
- Spatially-related properties,
- Reference system,
- Expansion,
- Content,
- Application schema,
- Portrayal catalogue,
- Distribution,
- Conditions of use.

In accordance with ISO, quality and metadata can be detailed for a database (collection of logically associated objects) for reporting groups (part quantities of a database) and for individual objects.

The common AFIS-ALKIS-ATKIS metadata catalogue is described in Chapter 9.

3.6 Feature catalogue

The structure of the feature catalogue is specified by ISO 19110 *Feature Cataloguing Methodology*. It is also possible to use object orientation to describe the methods in the feature catalogue. The common application schema expands these structures within the

AAA Catalogue package, which are also required for the AFIS, ALKIS and ATKIS applications.

For simpler implementation, catalogues are versioned and exchanged as complete items only.

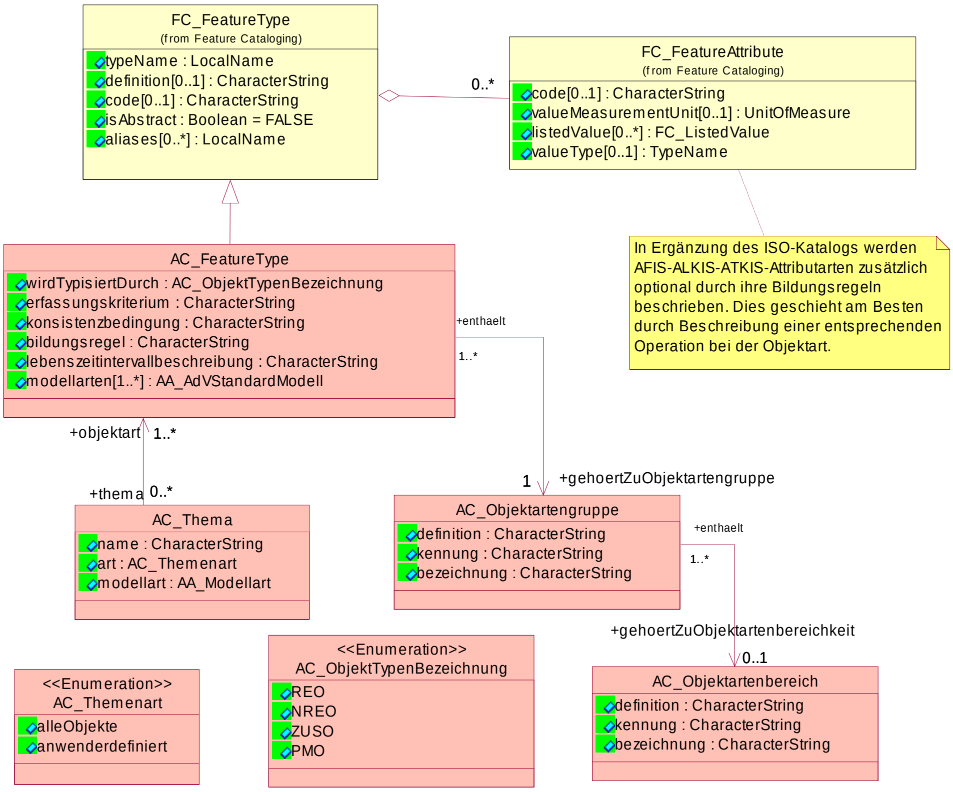


Figure 3-17: Expansion of standardised structure of the feature catalogue

3.7 Processes, operations and methods

3.7.1 Fundamental principles

The official surveying and mapping remit includes the collection, qualification, management (initial set-up, updating), use and transfer of data. Each of these tasks is expressed on one or several processes, i.e. collection, qualification, management, use and transfer processes.

The geoinformation of official surveying and mapping consists of original primary database data and temporary data inventories in the collection, updating, output and transfer data.

Figure 3-17 shows the processes and data for the geoinformation for official surveying and mapping. The database parts to be modelled as part of the AdV Project ”Modelling geoinformation of official surveying and mapping” are bordered by a dotted line.

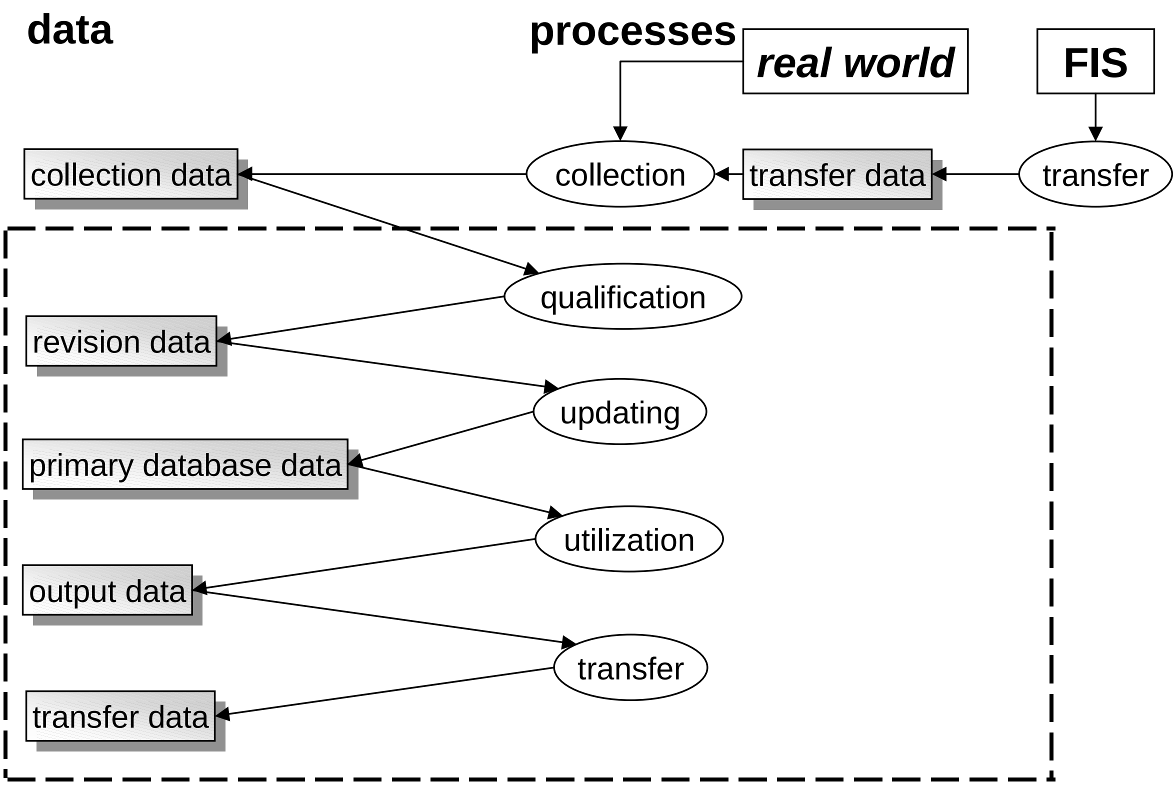


Figure 3-18: Processes and data for Geoinformation of Official Surveying and Mapping

One process comprises several modularly arranged methods, which can be grouped into operations and sub-divided by technical area. The following terms are used to describe the process (operations and methods):

- Methods as a constituent of the UML classes
- Textual description of processing steps
- Sequence diagrams
  The operations are graphically represented in accordance with the UML notation in sequence diagrams
- Filter encoding expressions (see Section 10.2.2) for describing the selection and evaluation functionality when creating standard outputs (utilisation process).

3.7.2 Business process

Business operations regulate the procedures used in official surveying and mapping and one of their constituents is the dynamic part of an application schema for performing qualification, management, use and data transfer tasks in the form of modularly arranged operations. Business process modelling is not part of the AAA application schema.

The business process carries out selected collection, qualification, management, use and transfer operations. (Figure 3-18).

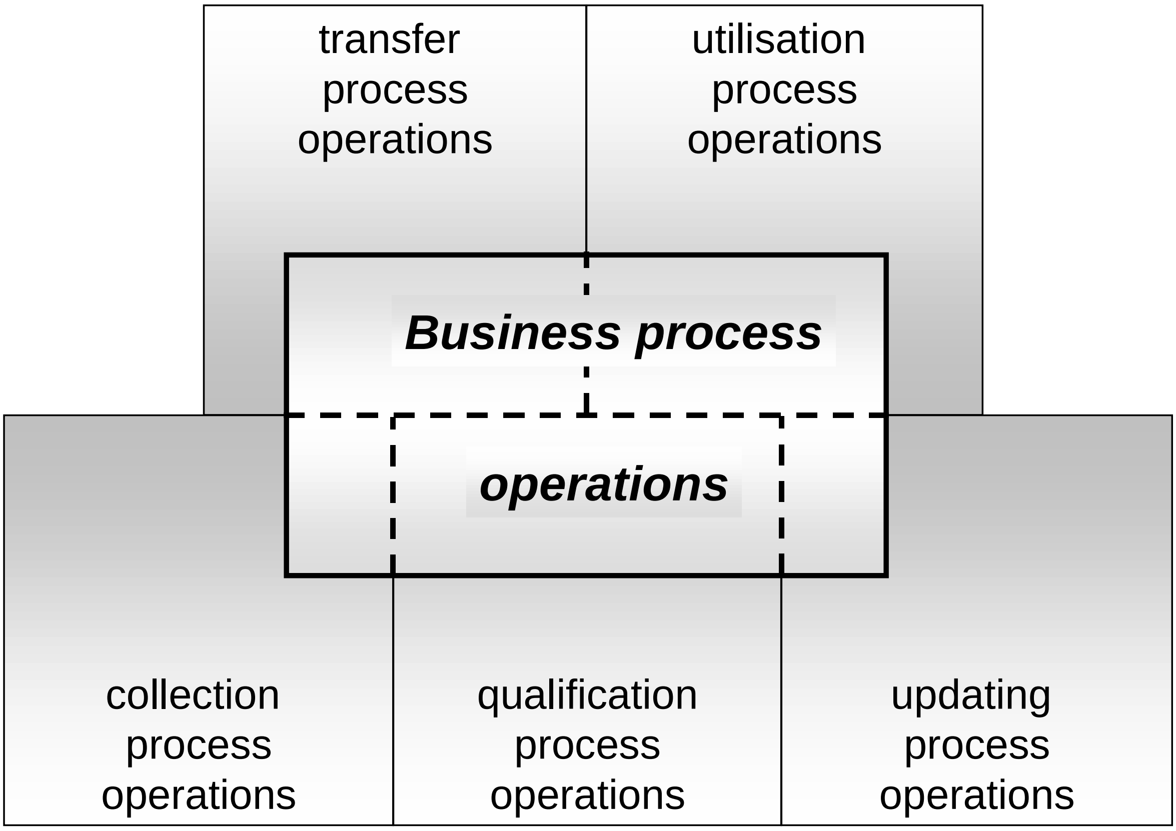


Figure 3-19: Operations of a Business Process

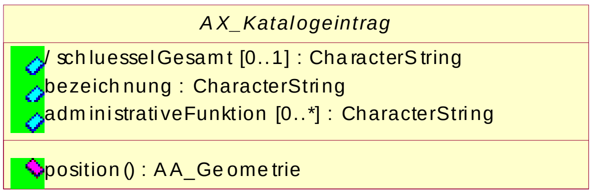
3.7.3 Operation and method

For a complete application description, operations and methods must be defined that set the data in functional dependencies and define the dynamic behaviour of the application.

A operation contains the representation of processing steps for qualification, management, use and transfer, in which reference is made to various methods. The operation is part of a business process. Operations for the utilisation process can be assigned to individual output objects. For example, the “Create land parcel certificate” operation creates the “Land parcel certificate” output feature type within the ALKIS utilisation processes.

A method describes the behaviour of an object and consists of a sequence of instructions. The object receives the impulse to do this through a message, which is triggered through user inputs or through methods of other objects (input parameters). The result of the methods is prepared in the form of output parameters. Methods are defined in relation to the object and are part of a class within the UML model.

Here an example of ALKIS modelling:



The 'position()' method determines the area of the object, derived from the object geometries that relate to the catalogue entry.

3.7.4 Processes of the AFIS-ALKIS-ATKIS application schema

A process transfers a source database to a target database. The utilisation process, for example, transfers the primary database data into temporary output data.

To control the various processes, special classes are formed (see also Section 10.2), which contain control parameters for a process sequence, e.g. "AX_utilisation request" in the utilisation process of the ALKIS application schema.

3.7.4.1 Collection process

Source data are examined using the known geodetic measuring and reconnaissance methods in the real world or recorded from cartographic representations and other documents. The target data of the collection process are the object-structured collection data, which form a basis for updating the official geoinformation.

3.7.4.2 Qualification process

In the qualification process, the digital, object-structured collection data are transferred to updating data following qualification. This is a method of quality assurance and ensures that the updating data satisfies the quality requirements.

The target data of the qualification process are the updating data.

3.7.4.3 Updating process

The updating process combines the initial set-up and updating of the geoinformation, where initial set-up can be considered a special case of updating. The updating process involves the updating data (data and metadata) being transferred to the database by applying suitable methods.

The target data of the updating process are the primary database data.

The functionalities required for set-up and updating are described as part of the data exchange interface in 10.2, any other implicit functions of a management system are described in Section 10.4. The conceptual technical model for updating of ALKIS and the exact procedures during update processing are contained in the documentation on feature type “AX_revision request”. A sequence diagram also shows an example illustration to describe the “AX_revision request”.

3.7.4.4 Utilisation process

Utilisation processes transfer primary database data into output data in accordance with the technical definitions:

- in the form of primary database records for universal further processing at the user,
- as prepared primary database data with specified content and the standard appearance of official surveying and cadastre (presentation outputs, evaluations etc.) and also
- as change data following updating (user-related update of primary database – NBA).

An output can contain database objects and temporary objects.

Temporary feature types (TOA) are created (e.g. TOA land parcel_land register) for structuring the output data and element details that cannot be taken from the attribute types of the database. Temporary feature types are data types rather than AA_objects. They have no identifier and no lifetime interval. They are not therefore included in the database.

The temporary process feature type “AX_utilisation request” of the application schema contains essential details for controlling the utilisation process such as output scope, application number, initiation type, utilisation parameters, output name and/or is generated at the start of the utilisation process. Through the “utilisation parameter” attribute type, the parameters required for the cost and fee calculation performed outside of the geoinformation of official surveying and mapping are provided. The other temporary feature types required for an output are created from the stock feature types by methods within the utilisation process.

The temporary feature types, specifically the temporary output feature types are modelled so as to avoid relations within an output. One exception is the “utilisation request” temporary feature type, which is to be linked to the associated output by a relation, because the utilisation request is to be returned to the request manager at the end of the utilisation process.

Output feature types can also be presented in compliance with the potrayal catalogue, depending on requirement. The links and the information flow between the feature catalogue, the output catalogue, the portrayal catalogue and the outputs can be taken from the schema shown below (Figure 3-20).

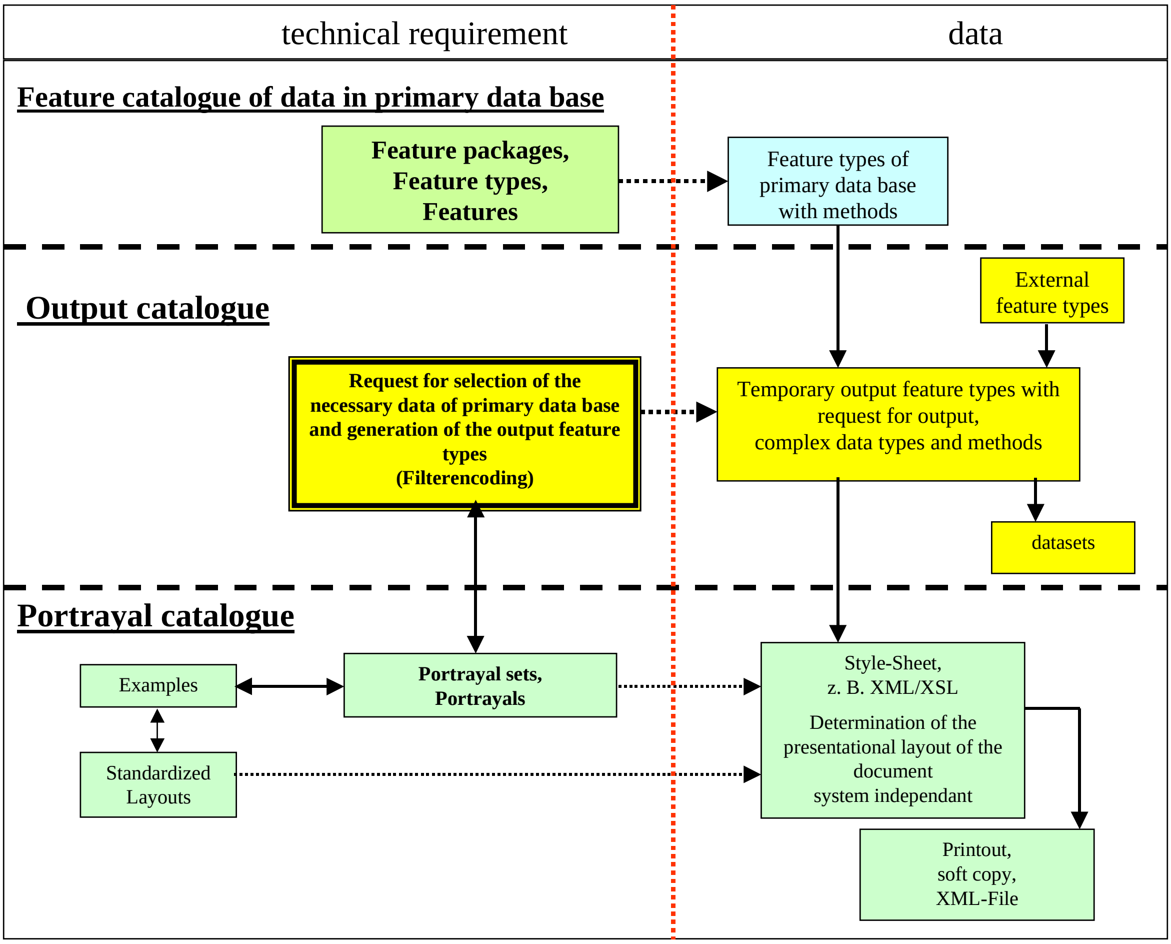


Figure 3-20: ALKIS output schema

Accordingly, for a presentation output for example, the feature types of the primary database data are prepared using a defined enquiry sequence and developed to create a temporary output feature type. Taking into account the necessary portrayal and presentation layout, an output then appears on screen and/or is printed.

3.7.4.5 Transfer process

Transfer processes occur on transfer of third party data in the form of updating data and on submitting output data to customers. Transfer processes for data acceptance receive outputs from third party systems, including transfer functions in the form of transfer data. Transfer processes for data output supplement output data with transfer functions and generate from them transfer data for third party systems.

4 Catalogues of the basic schema and the versioning schema

The catalogue of the basic schema and the versioning schema contains documentation on the schema classes described in the UML as readable text document.

The catalogue of the basic schema is available as a HTML-Document and as a PDF-File.

The catalogue of the versioning schema is also available as a HTML-Document and as a PDF-File.

5 Theme-specific applications of the Basic Schema

5.1 The common AFIS-ALKIS-ATKIS technical schema

The theme-specific requirements from AFIS, ALKIS and ATKIS resulted in the development of the common AFIS-ALKIS-ATKIS technical schema. This common schema is available with the Basic Schema and the used ISO standards as a Rational Rose file. The required files can be found in the following ZIP archive.

AAA.zip

5.2 Explanations on AFIS

The explanations for the AFIS-application are available as a PDF-file.

5.3 Explanations on ALKIS

In this chapter, the structure and content of the ALKIS-application are explained. Almost all object types (features) are commented and examples are shown.

The explanations for the ALKIS-application are available as a PDF-file

5.4 Explanations on ATKIS Digital Landscape Models

In this chapter, the structure and content of the ATKIS-DLM-application are explained. For special feature types detailed explanation and examples are provided. At the moment, exlpanations are only available for the ATKIS Base-DLM.

The explanations for the ATKIS-Base-DLM-application are available as a PDF-file

5.5 Explanations on ATKIS Digital Terrain Models

5.6 Explanations on ATKIS Digital Topographic Maps

6 AFIS catalogue documents

6.1 AFIS feature catalogue

The AFIS feature catalogue is available as a PDF-File and an HTML-Document.

6.2 AFIS output catalogue

The AFIS output catalogue contains the output feature types derived from the UML model. The examples are annotated with explanations. The output catalogue is available as a PDF-File

7 ALKIS catalogue documents

7.1 ALKIS feature catalogue

The ALKIS feature catalogue consists of title page, table of contents and the following sections:

- Preliminary remarks (general explanations) – Part A.
- The feature catalogue (OK) derived from the Rational Rose Model - Section B. This Section is available as an HTML-Document.
- These two Sections are grouped into a PDF-File.

7.2 Processes in ALKIS

- The “Processes in ALKIS“ document contains the operators for controlling the processes in ALKIS, the output feature types with the complex data types and also an example for the derivation of a land parcel certificate from the ALKIS database using filter encoding. The document is available as a PDF-File.

7.3 ALKIS portrayal catalogue

The ALKIS portrayal catalogue consists of:

- title page, table of contents and the preliminary remarks – Part A.
- the tabulated portrayal catalogue – Section B – as a PDF-File.

8 ATKIS catalogue documents

8.1 ATKIS feature catalogues

8.1.1 ATKIS feature catalogues for the Basic Digital Landscape Model (basic ATKIS-OK)

The ATKIS feature catalogue consists of title page, table of contents and the following sections:

- Preliminary remarks (general explanations) – Part A.
- The feature catalogue (OK) derived from the Rational Rose Model - Section B.
  This Section is available as an HTML-Document.
- These two Sections are grouped into a PDF-File.

8.1.2 ATKIS feature catalogue for the Digital Landscape Model 1:50,000 (ATKIS-OK50)

8.1.3 ATKIS feature catalogue for the Digital Landscape Model 1:250,000 (ATKIS-OK250)

8.1.4 ATKIS feature catalogue for the Digital Landscape Model 1:1,000,000 (ATKIS-OK1000)

8.1.5 ATKIS feature catalogue for the Digital Terrain Model

8.2 ATKIS portrayal catalogue

8.2.1 ATKIS portrayal catalogue for the Digital Topographic Map 1:10,000 (ATKIS-SK10)

8.2.2 ATKIS portrayal catalogue for the Digital Topographic Map 1:25,000 (ATKIS-SK25)

8.2.3 ATKIS portrayal catalogue for the Digital Topographic Map 1:50,000 (ATKIS-SK50)

8.2.4 ATKIS portrayal catalogue for the Digital Topographic Map 1:100,000 (ATKIS-SK100)

8.2.5 ATKIS portrayal catalogue for the Digital Topographic Map 1:250,000 (ATKIS-SK250)

8.2.6 ATKIS portrayal catalogue for the Digital Topographic Map 1:1,000,000 (ATKIS-SK1000)

9 Metadata catalogue

The AFIS-ALKIS-ATKIS metadata catalogue consists of title page, table of contents and the following sections:

- AFIS-ALKIS-ATKIS metadata catalogue with selected metadata elements in accordance with draft ISO/FDIS 19115:2002; Metadatacatalogue - Part A
- Examples – Part B
-

10 The external model, data exchange

Chapter 2 describes the bases and correlations for the geoinformation to be described with this documentation. The specified reference model also represents the requirement for data exchange. Where it is necessary to define the data exchange as AdV standard, this Chapter contains the stipulations for the data exchange interfaces to be used. The XML schema files cited below can be found in this ZIP-Archiv.

10.1 Standards-based data exchange interface (NAS)

The standards-based data exchange interface (NAS) is used when it is necessary to exchange geoinformation that has been modelled as object data on the basis of the common AFIS-ALKIS-ATKIS application schema. This can relate both to information that has the same structure as the stored data inventories, including the additional data (presentation objects, map geometry objects, see Chapter 2) and also to information from derived views on these data inventories (e.g. output feature types, see Section 7.2), but not to data inventories for which the object reference is completely lost (e.g. purely graphically structured data) or data that is to be defined according to a different basic schema (e.g. DXF data).

Accordingly, the NAS is used wherever the application emphasis is on

- the originality of the data,
- the full evaluation capability and
- the feature-specific updating,

in line with the user's requirement.

10.1.1 Norms and standards

The standards AFIS, ALKIS and ATKIS of the AdV are described in this document in conceptual format on the basis of ISO 19109 *Rules for Application Schema*. This means specifically:

- Modelling in UML with the software tool Rational Rose
- Compliance with the regulations of ISO 19103 for the use of UML
- Use of ISO 19107 (and therefore by implication ISO 19111), ISO 19115
- Automated derivation and mapping of feature catalogues in acc. with ISO 19110

Automated derivation of the interface for exchange of AFIS, AKIS and ATKIS objects, the NAS, completes this picture.

For this purpose, ISO 19118 *Encoding* defines a framework document for the creation of what are referred to as *Encoding Rules*, to derive interface definitions for data exchange from a UML application schema. The de-jure standard also describes in an informative Appendix, special *Encoding Rules* for the creation of XML schema definitions. The variability permitted for mapping UML to XML schemas does, however, result in the ability of ISO 19100 basic norms to be converted in different ways. A stipulation by the AdV would result in AdV-specific interfaces and thus, the purpose of the de-jure standards, namely to achieve inter-operability, would not be fulfilled. There are currently no standardised XML schemas for the basic ISO-standards.

Besides the official de-jure standards for defining interfaces definitions, there is a de-facto standard, based on the *Geography Markup Language* (GML) of the *OpenGIS Consortium* (OGC), for encoding geoinformation, which is also provided for standardization in the ISO 19100 series as ISO 19136. This provides a fixed XML schema for basic constructs/samples (features, geometry, associations, etc.). GML is gradually achieving market acceptance.

In accordance with the fundamental principles cited in Chapter 3.1, the AdV uses the new development of AFIS, ALKIS and ATKIS to pursue the objective of creating the basis for a common, unified and interdisciplinary use of geodata. In this sense, existing or foreseeable standard functionalities of application software should be used wherever possible. One example is the NAS described in this Chapter. Use of AdV-specific solutions is largely avoided. Due to the current status of international standardization in the field of metadata and the selection expressions for GML-data , however, this is currently only possible to a certain degree.

For these reasons, the AdV has decided for the following procedure:

- The framework document for Encoding Rules defined in ISO 19118, Section 8 is applied for the NAS (Level-1 – Conformity with ISO 19118).
- "NAS Encoding Rules" according to ISO 19118, Section 8 are defined and documented. These "NAS Encoding Rules" map the conceptual AFIS-ALKIS-ATKIS model on a GML-3.0 application schema.
- The "NAS Encoding Rules" are stipulated in conformity with GML 3.0 and furthermore as simple as possible and as far as possible in conformity with ISO 19118 "XML Encoding Rules".
- An automatic derivation of NAS is supported, which means that the XML schema definitions of NAS can be derived using the "NAS Encoding Rules", the UML application schema and additional regulations formally described in the form of control parameters.

It is important to note that the stable conceptual model is fully described in the UML application schema. Future adaptations to the IT/GI Mainstream will also become necessary for depiction on specific implementation models (e.g. XML representations).

Besides the stated de-jure standards of the ISO 19100 series of standards, the following documents are used to define NAS:

- UML 1.3:1999, *Unified Modeling Language (UML)*, Object Management Group (OMG), http://www.omg.org/
- XML 1.0:1998, *Extensible Markup Language (XML)*, W3C Recommendation, 6 October 2000, http://www.w3.org/TR/2000/REC-xml-20001006
- XML Schema Part 1: *Structure* - W3C Recommendation, 2nd May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/
- XML Schema Part 2: *Data types* - W3C Recommendation, 2nd May 2001, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/
- XLink *XML Linking Language* (XLink) Version 1.0, W3C Recommendation 27 June 2001, http://www.w3.org/TR/2000/REC-xlink-20010627/
- Geography Markup Language (GML) 3.0
  Open GIS Consortium, 2003
  Online
- Web Feature Service 1.0
  Open GIS Consortium, 2002
  Online
- Filter Encoding 1.0
  Open GIS Consortium, 2002
  Online

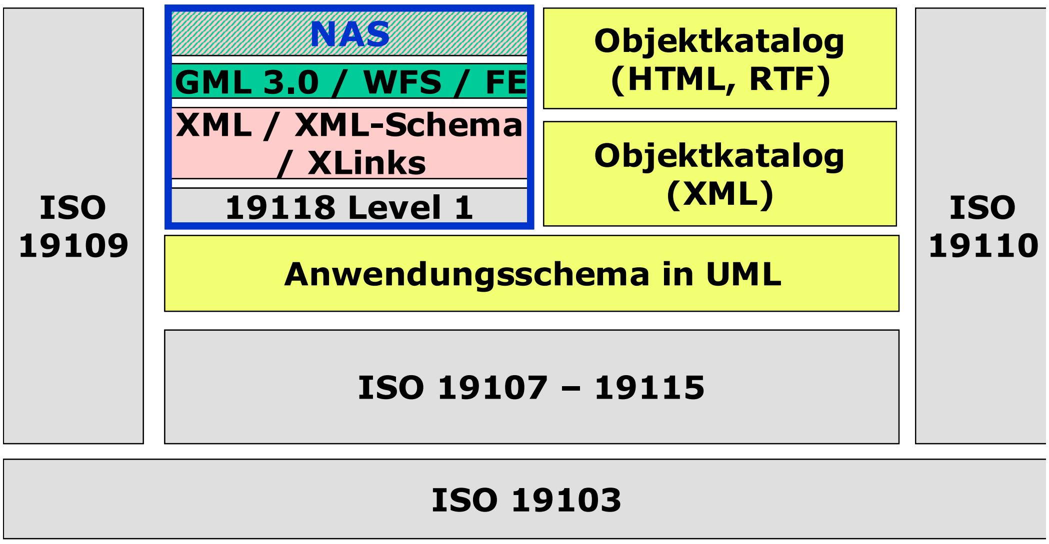


Figure 10-1: Embedding the NAS in de-jure standards and de-facto standards

10.1.2 Encoding process

ISO 19118 describes in general terms the encoding and decoding process to be carried out as follows:

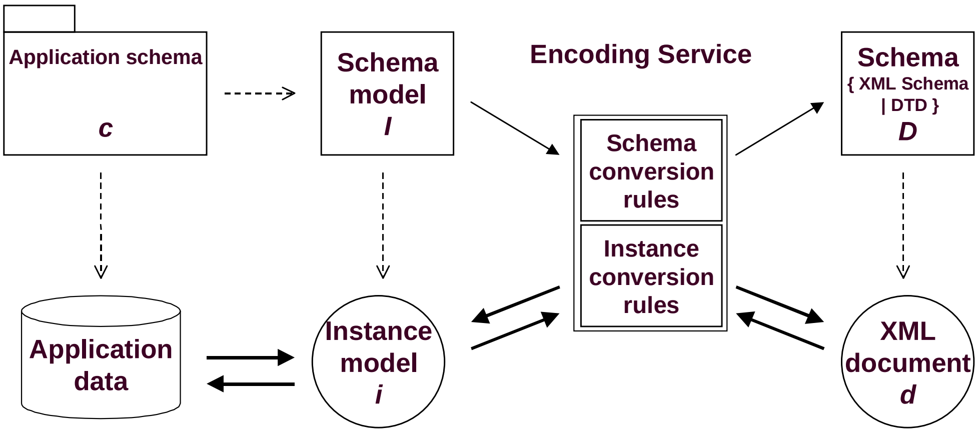


Figure 10-2: XML-based encoding regulations in accordance with ISO 19118

The process is based on the following framework conditions:

- A formally (e.g. in UML) described application schema exists.
- The information of the UML application schema is transferred to an XML schema file on the basis of schema conversion rules and where applicable control instructions.

- In the same way, the application data based on the application schema (object instance) are transferred to an XML file that has the same structure as the definitions of the SML schema using instance conversion rules.

Within the context of NAS, the UML schema is converted into the XML schema by a software script within the *Rational Rose* modelling software. A satisfactory realisation of a manufacturer-independent solution (transfer of the UML schema into XMI or similar) does not currently appear possible. The control parameters are also defined in *Rational Rose* (see 10.1.3.4). This gives rise to the following sequence for automatic derivation of the NAS schemas:

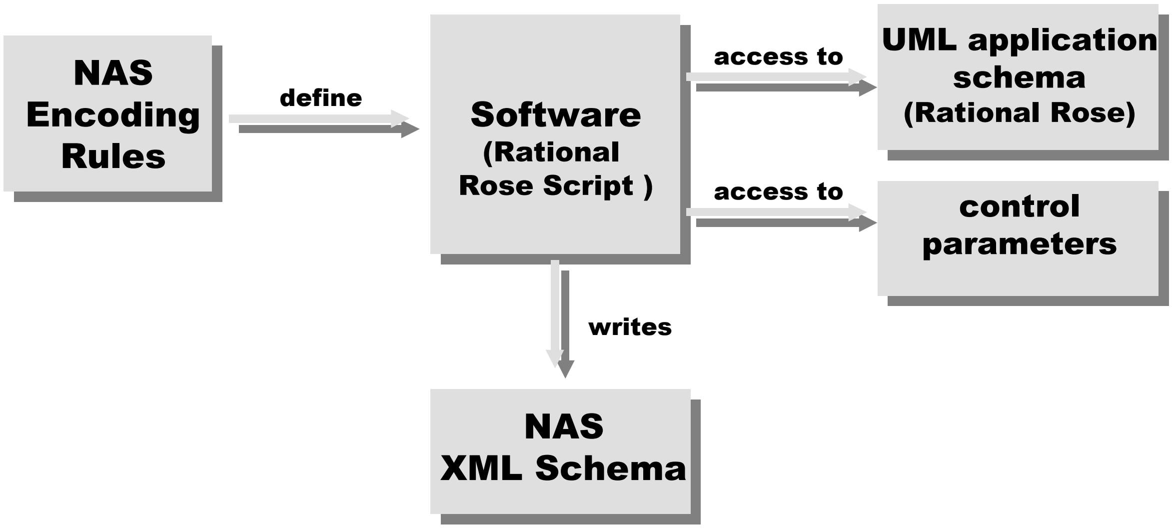


Figure 10-3: Automatic derivation of the NAS schemas

Accordingly, the NAS structure is based on the AAA-basic schema described in Chapter 3, the application-specific sub-schemas for AFIS, ALKIS and ATKIS (see Chapter 5) and the corresponding control parameters.

10.1.3 NAS Encoding Rules

The "NAS Encoding Rules" are described below. The structure fulfils the requirements of ISO 19118, Section 8 and is based on ISO 19118 Annex A for reasons of simple comparability.

Section 8 of ISO 19118 specifies the requirements for Encoding Rules. An Encoding Rule describes the mapping rules by which data from an input data structure (instances in accordance with the AAA application schema in Rational Rose) can be transferred to an output data structure (XML-file in accordance with NAS). An Encoding Rule covers the following themes:

- Requirements
    1. Application schema
    2. Character set and supported languages
    3. *Exchange metadata*
    4. Identifiers
    5. Update mechanisms
- Input data structure
    1. Instance schema
    2. Relationship between instance schema and application schema
- Output data structure
- Presentation regulations
- Examples

10.1.3.1 Requirements

Application schema

The AAA-basis schema and AAA-thematic schema were developed on the basis of the regulations for application schemas from ISO 19103 and ISO 19109. The corresponding regulations from ISO 19118 A.2.1 also apply for the "NAS Encoding Rules".

Both the AAA-thematic schema and the AAA-basic schema on which it is based are transferred to separate XML schema definitions using the NAS Encoding Rules described here, where the thematic schema in XML schema is based on the basic schema in the XML schema, too. The basic schema must be conclusive and must hold no references to the thematic schema.

The two schemas must be carried in the UML model in separate *Packages*. Both packages must carry the *Stereotype* <<Application Schema>>.

The use of inheritance is permitted only for objects with identity (*Stereotypes* <<Feature>>, <<Type>> and without *Stereotype*). This is mainly to ensure that the XML schema structure of NAS is created as simply as possible. Hereditary is avoided for *DataTypes*, so that it is not necessary to apply ISO 19118 A.5.2.9. In the case of contained elements – *DataTypes* – it is not necessary to define the associated *choice* elements, which makes the XML schema easier to evaluate.

The operations that can be run using NAS are also modelled in UML and automatically transferred to XML definitions. For this purpose, request object classes are modelled with the *Stereotype* <<Request>> and response object classes with the *Stereotype* <<Response>> (see Section 10.2). The object classes must be in a *Package* with the *Stereotype* <<Application Schema>>. On representation in the NAS, each of these object classes is transferred to an XML root element in its own XSD file. All classes for one operation, i.e. a request or a response must be located with that operation in one package. Consistency with ISO 19119 *Services* must be ensured when modelling operations, i.e. requests and responses.

All references to classes from the fundamental ISO schemas must be defined in concrete terms in order to be correctly represented in NAS through control parameters. Either the ISO class is to be marked for export via control parameter "ConsiderISOType" or the attribute and/or association in the AAA-basic schema or thematic schema must be represented through an alternative method (via control parameters "Suppress", "ReplacingElement" or "ReferencedElement").

A central location is held for the description of NAS operations. Each provider of NAS operations, i.e. each location that manages a primary database, should supply the relevant *Service Metadata* in accordance with ISO 19119 *Services*, Sub-section 6.6. The name of the *Service Interface* (property "SV_ServiceIdentification.serviceType") is "NAS" with the "AdV" *nameSpace*.

Character set and supported languages

As specified in ISO 19118 A.2.3, the *Universal Character Set* (UCS) of "ISO-10646-1" should in principle be used as the character reserve. This is identical to the *Unicode Character Repertoire.*

"UTF-8" (*UTF = UCS Transformation Format*) should uniform be used as *Character Encoding* for NAS data. "UTF-8" is also the standard value in XML, should an encoding detail be missing.

The language is German ("de") or Sorbian (lower Sorbian and/or upper Sorbian).

Exchange Metadata

As part of modelling the requests and responses, the required exchange metadata is modelled and transferred by automatic conversion to the XML schema.

Identifiers

In NAS, identifiers are defined only at the level of the feature, i.e. in all XML-elements representing the types that are a sub-class of AA_object. For these types, the identifiers must always be indicated (with the exception of the cases defined below). Identifiers for all other elements are read over and not observed.

The identifiers for features must always be understood within the meaning of UUIDs, i.e. unique within the "*AFIS-ALKIS-ATKIS" Application Domain*".

The AAA-identifier consists of 16 characters. The structure is described in Chapter 3.3.7.

References to other objects are always represented as an XLink. Within the NAS, references to other AAA-objects must, without exception, be expressed through URNs. *Uniform Resource Names* (URNs) are persistent identifiers independent of memory location, in which *namespaces* can be indicated. AAA-identifiers are located in *URN-namespace* "adv:oid".

Example: "urn:adv:oid:DENWAL1212345678".

The identifier is supplemented in the following cases by the time at the start of life of a feature:

- In the utilisation response, if more than one version of an object is included within one NAS-file, in the attribute gml:id of the object (see chapter 10.2.3).
- In an update request for update in the attribute fid of the filter expression (see chapter 10.2.3).

For references to other objects (e.g. in associations), however, the identifier is always used without the time.

Update mechanism

An update mechanism as defined by ISO 19118, Section 8 is supported (see Item 10.1.6)

10.1.3.2 Input data structure

The instance schema is used with the following changes from ISO 19118, Annex A.3.

Given the aforementioned stipulations for the identifier, identifiers must always be indicated in the IM_object in "id". A redundant repetition in "uuid" is not permitted and this attribute is always missing.

Table: Overview of stipulations for representing UML concepts in the instance schema

UML	Instance schema
<<Interface>>	see ISO 19118 A.3.2
<<BasicType>>	see ISO 19118 A.3.2
<<DataType>>	see ISO 19118 A.3.2
<<Request>>	IM_StructuredValue
<<Response>>	IM_StructuredValue
<<Union>>	see ISO 19118 A.3.2
<<Enumeration>>	see ISO 19118 A.3.2
<<CodeList>>	see ISO 19118 A.3.2
<<ExternalCodeList>>	as <<CodeList>>
<<Feature>>	IM_Object
<<Type>> or no stereotype	see ISO 19118 A.3.2
Abstract class	see ISO 19118 A.3.2
Derived classes	see ISO 19118 A.3.2
Attributes	see ISO 19118 A.3.2
Association	see ISO 19118 A.3.2
Aggregation	as association
Composition	see ISO 19118 A.3.2
Methods	see ISO 19118 A.3.2
OCL constraints	see ISO 19118 A.3.2

Only properties with global visibility, no attributes or relations that are identified as "*private*" or "*protected*" are represented.

Derived attributes (“*derived*“*)* are represented in NAS by default, the same applies for class attributes.

For all other items not mentioned above, the provisions in ISO 19118, Annex A.3.2 are applied.

10.1.3.3 Output data structure

The instance model is represented through NAS Encoding Rules in an XML document, in conformity with ISO 19118 A.4. The result of the representation is valid and relates to the corresponding, automatically generated XML schema.

10.1.3.4 Schema encoding rules

XML Schema

Each time a UML-schema is transferred, a package tagged with the <<Application Schema>> *stereotype* is mapped into an XML schema (XSD file). Exactly one package is transferred to one XSD file.

This applies for both the AAA-thematic schema and the AAA-basic schema on which it is based. Both are transferred with the described NAS Encoding Rules into separate XSD-files named "AAA-basicschema.xsd" and "AAA-technical schema.xsd" respectively, where the thematic schema is also based on the basic schema.

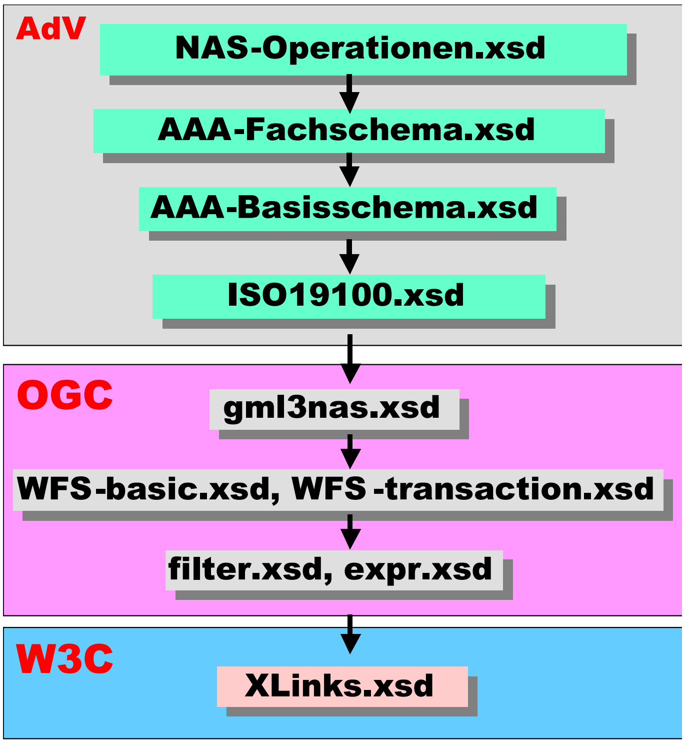


Figure 10-4: Basic structure of the central XML schemas of NAS

Thus, the basic schema and thematic schema are also separated in the XML schema definitions. Both schemas are located in *XML-namespace* "xmlns:adv=http://www.adv-online.de/namepsaces/adv/gid3.0" . One essential reason for this separation is to enable other technical areas outside of the surveying authority to develop their own thematic schemas based on the AAA-basic schema. These areas can model their own thematic schemas and represent them according to the XMl schema using the NAS Encoding Rules. For this application, the file "AAA-technical schema.xsd" is to be replaced by a suitable alternative. These application schemas must be located in a namespace different to the one stated above, the AdV reserves the right to use the AdV namespace.

Based on the AAA-thematic schema, XSD files for describing the requests and responses of each operation are defined for the individual operations to be run with NAS (see Item 10.1.6).

In principle, the statements in ISO 19118 A.5.1 apply.

An XSD file ISO19100.xsd is generated for the classes taken directly from the ISO schemas. Because this is an AdV-specific conversion (ISO has not previously stipulated its own XMl schema definitions that could be used), the elements and types are located in *adv-namespace*.

Basisdata types

The representation of basisdata types (<<BasicType>>) in accordance with ISO 19118 A.5.2.1 remains unchanged for NAS.

The following applies for the basisdata types currently used in the UML model:

Table: Representation of basisdata types

Basisdata type in UML	Data type in NAS
Integer	xs:integer
Number	xs:decimal
Decimal	xs:decimal
Real	xs:decimal
CharacterString	xs:integer
Date	xs:date
DateTime	xs:timeInstant
Boolean	xs:boolean

Data types

Data types (<<DataType>>) are converted in the same way as ISO 19118 A.5.2.2. It must be ensured that in NAS, attributes and associates different to those described in ISO 19118 A.5.3 are converted (see below).

Enumeration

Enumerations (<<Enumeration>>) are in principle converted as described in ISO 19118 A.5.2.3.

Code lists

Code lists (<<CodeList>>) are converted in the same way as enumerations (ISO 19118 A.5.2.3). The code –if present-, not the value, is encoded.

Note: This means that also code lists may also only be expanded by the AdV and a change results in a new version of the NAS.

External code lists

External code lists (<<ExternalCodeList>>) are converted as a GML-3.0-Dictionary. Accordingly, rather than in the output schema, they appear in *dictionaries* format in an "external" XML-file. This is advantageous since expansions to these code lists do not result in a new NAS version.

In the NAS-file <name> is encoded, not the gml:id or the <description>.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<Dictionary gml:id="AAAExternalCodeLists"
      xmlns=http://www.opengis.net/gml
      xmlns:gml="http://www.opengis.net/gml"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.opengis.net/gml
            gml/3.00/base/dictionary.xsd">
      <description>Externe Codelisten von AFIS-ALKIS-ATKIS.</description>
      <name>AAA Codelisten</name>
      <dictionaryEntry>
            <DefinitionCollection gml:id="AA_Anlassart">
            <description>Stand: 2003-03-10</description>
            <name>AA_Anlassart</name>
            <definitionMember>
                  <Definition gml:id="AA_Anlassart_1">
                  <description>Nach Quellenlage nicht zu spezifizieren</description>
                  <name codeSpace="adv:AA_Anlassart">9998</name>
                  </Definition>
            </definitionMember>
            <definitionMember>
                  <Definition gml:id="AA_Anlassart_2">
                  <description>Sonstiges</description>
                  <name codeSpace="adv:AA_Anlassart">9999</name>
                  </Definition>
            </definitionMember>
            <definitionMember>
                  <Definition gml:id="AA_Anlassart_3">
                  <description>Flurstücksdaten fortführen</description>
                  <name codeSpace="adv:AA_Anlassart">010000</name>
                  </Definition>
            </definitionMember>
            <!-- ... -->
            <definitionMember>
                  <Definition gml:id="AA_Anlassart_127">
                  <description>Änderung der Anschrift</description>
                  <name codeSpace="adv:AA_Anlassart">090300</name>
                  </Definition>
            </definitionMember>
            <definitionMember>
                  <Definition gml:id="AA_Anlassart_128">
                  <description>Änderung der Personendaten</description>
                  <name codeSpace="adv:AA_Anlassart">090400</name>
                  </Definition>
            </definitionMember>
            </DefinitionCollection>
      </dictionaryEntry>
      <!-- ... -->
      <dictionaryEntry>
            <DefinitionCollection gml:id="AX_CRS">
            <description>Stand: 2003-03-10</description>
            <name>AX_CRS</name>
            </DefinitionCollection>
      </dictionaryEntry>
</Dictionary>
```

Union

Union data types (<<Union>>) are converted as described in ISO 19118 A.5.2.5.

Objects

In conformity with the provisions in GML 3.0, Chapter 8., Feature types (<<Feature>> and <<Type>> and also classes without *Stereotype*) are converted as *complexType;* the name is derived from the class name plus the suffix "Type". Through the XML schema "Extension mechanism", the *complexTypes* "inherit" either directly or indirectly from gml:AbstractFeatureType (and therefore also the mechanism for describing identifiers).

For AFIS-ALKIS-ATKIS objects, the AAA-identifiers are always indicated in the "gml:id" attribute, e.g.:

```
<AX_Flurstueck gml:id="DEBY0000F0000001">
  ...
</AX_Flurstueck>
```

Set<T> and Sequence<T>

Properties that use these template classes as type are converted as GML array properties.

Inheritance / abstract classes

Within an XML-schema (i.e. AAA-basic schema and AAA-thematic schema), the properties are propagated into page classes by the "*copy down*" mechanism described in ISO 19118.

Therefore, abstract classes are not usually represented explicitly as *complexType* (instead, the properties are propagated in the instance-capable classes). Exceptions are the abstract classes of the basic schema, from which the features of the thematic schema inherit. These are defined in the AAA-basic schema (tagged with attribute 'abstract="true"', see ISO 19118 A.5.2.10). The abstract classes to be accepted into NAS are to be indicated for automatic NAS derivation though the relevant control parameters.

In the AAA-thematic schema, the properties of the abstract top classes are not accepted directly from the basic schema through the "*copy down*" mechanism. Instead these classes are expanded by the "*XML-schema extensions*". The result is an extremely flat inheritance structure in the XML schema. This regulation for converting hereditary lies in the intermediate area of the two alternatives defined in ISO 19118 A.5.2.8 and uses elements from both.

Example:

```
<!-- aus AAA-Basisschema.XSD: -->

    <xs:complexType name="AA_ZUSOType" abstract="true">
          <xs:complexContent>
                <xs:extension base="gml:AbstractFeatureType">
                      <xs:sequence>
```

```
                              <xs:element name="lebenszeitintervall">
                                    <xs:complexType>
                                          <xs:sequence>
                                                <xs:element name="AA_Lebenszeitintervall"
                                                      type="adv:AA_LebenszeitintervallType"/>
                                          </xs:sequence>
                                    </xs:complexType>
                              </xs:element>
                              <xs:element name="modellart">
                                    <xs:complexType>
                                          <xs:sequence>
                                                <xs:element name="AA_Modellart"
                                                      type="adv:AA_ModellartType"
maxOccurs="unbounded"/>
                                          </xs:sequence>
                                    </xs:complexType>
                              </xs:element>
                              <xs:element name="anlass" minOccurs="0">
                                    <xs:complexType>
                                          <xs:sequence>
                                                <xs:element name="AA_Anlassart"
                                                      type="adv:AA_AnlassartType"
maxOccurs="unbounded"/>
                                          </xs:sequence>
                                    </xs:complexType>
                              </xs:element>
                              <xs:element name="zeigtAufExternes" minOccurs="0">
                                    <xs:complexType>
                                          <xs:sequence>
                                                <xs:element name="AA_Fachdatenverbindung"
                                                      type="adv:AA_FachdatenverbindungType"
                                                      maxOccurs="unbounded"/>
                                          </xs:sequence>
                                    </xs:complexType>
                              </xs:element>
                              <xs:element name="istTeilVon"
                                    type="gml:ReferenceType" minOccurs="0" maxOccurs="unbounded">
                                    <xs:annotation>
                                          <xs:appinfo>AA_ZUSO</xs:appinfo>
                                    </xs:annotation>
                              </xs:element>
                        </xs:sequence>
                  </xs:extension>
            </xs:complexContent>
      </xs:complexType>

  <!-- aus AAA-Fachschema.XSD: -->


      <xs:complexType name="AX_SicherungspunktType">
            <xs:complexContent>
                  <xs:extension base="adv:AA_ZUSOType">
                        <xs:sequence>
                              <xs:element name="punktkennung" type="xs:string" minOccurs="0"/>
                              <xs:element name="wirdVerwaltetVon" minOccurs="0">
                                    <xs:complexType>
                                          <xs:sequence>
                                                <xs:element name="AX_Dienststelle_Schluessel"
                                                      type="adv:AX_Dienststelle_SchluesselType"/>
                                          </xs:sequence>
                                    </xs:complexType>
                              </xs:element>
                              <xs:element name="sonstigeEigenschaft" type="xs:string"
                                    minOccurs="0" maxOccurs="unbounded"/>
                              <xs:element name="horizontfreiheit"
                                    type="adv:AX_Horizontfreiheit_NetzpunktType" minOccurs="0"/>
                              <xs:element name="relativeHoehe" type="gml:LengthType"
                                    minOccurs="0"/>
                              <xs:element name="vermarkung_Marke" type="adv:AX_MarkeType"/>
                        </xs:sequence>
                  </xs:extension>
            </xs:complexContent>
      </xs:complexType>
```

Attributes

The use of cardinalities is described in ISO 19118 A.5.3.1. All attributes are converted as elements (not as XML attributes).

The regulations in GML (see GML 3.0, Chapter 7.2 and Chapter 8) apply for the encoding of properties.

Associations

In accordance with the regulations for by-reference containing properties, associations are converted to GML. In principle, these associations should comply with ISO 19118 A.5.3.2. The essential difference lies in using "gml:AssociationAttributeGroup" instead of "IM_ObjectReference"; for references, the pre-defined GML-Type “gml:ReferencesType“ is always used in NAS. The target type of the association is indicated in an appinfo-annotation element for evalution through NAS parsers.

GML uses exclusively Xlink for relationships between objects. To represent the AAA schema, only a small selection of XLink options is required, specifically references to a different object via a URN. In NAS, only xlink:href-attributes are permitted so as to retain functional compatibility with the "uriref" variants of ISO 19118.

References to other objects are always made via the URN of the target object. The complete URN (in the name space "adv:oid") rather than a "#" reference must always be used, even if the object is described in the same document.

In NAS, only the associations identified in the application schema as a preferred direction are indicated. Where necessary, counter-relations must be generated and maintained by the implementation.

Note: The described representation regulation for associations does not guarantee that the target object from the expected feature type or in fact any valid target object exists. This test must be performed by the target system itself. For this purpose annotations were introduced. Example:

```
a) XML-Schema:

      <xs:complexType name="AX_FlurstueckType">
            <!-- ... -->
            <xs:element name="istGebucht" type="gml:ReferenceType">
                  <xs:annotation>
                        <xs:appinfo>AX_Buchungsstelle</xs:appinfo>
                  </xs:annotation>
            </xs:element>
```

```
            <!-- ... -->
      </xs:complexType>
```

b)XML-Datei:

```
<AX_Flurstueck gml:id="DEBY0000F0000002">
  ...
    <istGebucht xlink:type="simple"
                xlink:href="urn:adv:oid:DEBY0000B0000002">
  ...
</AX_Flurstueck>
```

Aggregations

To simplify and reduce the constructs required for NAS, aggregations such as associations are converted.

Compositions

In accordance with the regulations for by-value containing properties, associations are converted to GML.

Support for several languages.

No requirement is currently identified.

Element definitions

For each *Request* and each *Response*, a *Root Element* is defined in its own XSD file with the names of the object classes (see Item 10.1.6).

A global element is defined for each feature type (with 'substitutionGroup="gml:_Feature"'). The name of the element is the class name.

GML – and therefore also NAS – uses the "substitutionGroup" attribute to ensure that an XML element representing an object can also "behave" like a feature, i.e. can be located where an XML-element "_Feature" is expected. Example:

```
<xs:element
        name="AX_Flurstueck"
        type="adv:AX_FlurstueckType"
        substitutionGroup="gml:_Feature"/>
```

In all other cases, local elements are defined by default.

Pre-defined types and elements

The elements and types pre-defined in GML can be used in accordance with the regulations of GML. For operations (requests and responses) the elements and types of the Web Feature Server Spezifikation can also be used (see Item 10.1.6). The use of these types and elements in NAS is regulated by control parameters (see below).

The mapping of geometric modelling in the AAA-basic schema to GML within NAS is examined below.

Elements derived directly from the ISO schema can also be used (ISO19100.xsd). These currently lie exclusively in the area of the metadata.

Encoding of geometric properties of the basic schema in NAS

For NAS, it should be stipulated how the used classes defined in ISO-19100 standards are to be represented on the pre-defined types. This is done via control parameters (see below). The representation must show the information contained in the ISO model that is relevant for AFIS-ALKIS-ATKIS.

The orientation of lines is not encoded. Because the orientation of features (rivers etc.) is sometimes of interest, it has to be ensured that a) the line is captured in positive direction and b) the this orientation is maintained over all processes. Under this premiss the line-orientation within NAS is always positive and there is no need for a special tag.

For surface-rings the surface is, in accordance with ISO 19107, always to the left of the positively oriented lines.

In order to achieve the simplest possible structure for NAS, geometry is exchanged only on a redundant basis. Program systems reading in NAS data must detect any topology and/or common geometric usage relevant to this information. The entry barriers for using AFIS-ALKIS-ATKIS data are therefore kept as low as possible.

The "detection" of geometric division is defined as simply as possible through the following points.

Topological objects and objects with commonly used geometry can be assigned to themes. Topological relationships and common geometric usage are possible only within a theme. Themes are constrained to only one model-type, i.e. all features that take part in the theme must have the same model-type.

For two geometries to be identical, they must have identical definitions in a <Point> and/or <Curve>, an identical geometric pattern alone is insufficient on lines.

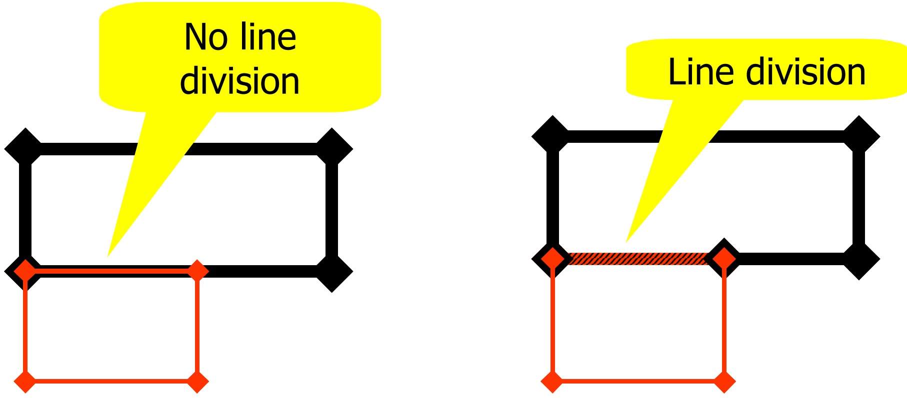


Figure 10-5: Explanation of line division

Identity on lines is always examined at the level of the "*GM_Curve*". Identity exists if all positions of the geometry definitions are identical in terms of position, sequence and used interpolation type. A sequence reversal is also permitted.

Two positions are identical if their distance is shorter than the required coordinate resolution. In AFIS-ALKIS-ATKIS, this is stipulated for metric position coordinates for 3 decimal places (mm). This stipulation applies independent of the actual accuracy of the coordinates.

The themes are represented in the NAS file as follows:

- The themes, and this applies to both "TS_theme" and "PointLineTheme", are (implicit) realisations of GM_Complex and ultimately an aggregation of geometric elements. Within the context of AFIS-ALKIS-ATKIS, two variants may occur. One is a complete form (type of theme declaration = "all objects", class themes) and the other is an instance-related form (type of theme declaration = "user-defined", instance themes).
- For the complete, class-related form all features of a feature class belong automatically to this theme. There is no choice. The explicit encoding of the theme at the instances is therefore not necessary.
- By the user-defined form instance-related geometric identities can be expressed, e.g. for the identity between a building and a parcel border. The names of the instance-themes which are used within a NAS-file have to be explicitly declared in the “Exchange Metadata”, i.e. they are modelled within the request- and response-objects. It is not necessary to limit this information to the in this particular NAS-file actually used themes. Together with the declaration of the instance-themes the kind

of geometry (identical points or lines, see above) decides, if there is an identity and if there should be a redundance-fee storing of the geometry or an explicit identity-association in the database. In case of a conflict between the theme declaration and the actual geometry, the theme declaration "wins", i.e. if there are identical geometries but no defined theme for the relevant features, no identity is assumed.

-

This can appear in an NAS file as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<AX_Fortfuehrungsauftrag
  xmlns:adv="http://www.adv-online.de/namespaces/adv/gid/3.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:gml="http://www.opengis.net/gml"
  xsi:schemaLocation="http://www.adv-online.de/namespaces/adv/NAS-
Operationen.xsd">

  <profilkennung>ii</profilkennung>
  <antragsnummer>4711</antragsnummer>
  <empfaenger>
    <AX_Empfaenger>
       <email>mailto:mustermann@foobar.de</email>
    </AX_Empfaenger>
  </empfaenger>
  <auftragsnummer>1174</auftragsnummer>
  <verarbeitungsart>1000</verarbeitungsart>


<verwendeteInstanzenthemen>Flurstücke und Gebäude</verwendeteInstanzenthemen>
    <verwendeteInstanzenthemen>Flurstücke und öffentlich-rechtliche
Festlegungen</verwendeteInstanzenthemen>
    <verwendeteInstanzenthemen>Flurstücke und tatsächliche
Nutzung</verwendeteInstanzenthemen>


...

</AX_Fortfuehrungsauftrag>
```

The theme-definitions are collected in an XML-file.

Encoding of the Point-Coverages in NAS

The encoding of the Point-Coverages in NAS the GML-encoding of the CV-classes of ISO 19123 is used:

- The class AD_PunktCoverage is defined as an extension of gml:MultiPointCoverageType (as an abstract XML schema type adv:AD_PunktCoverage).

- The class AD_GitterCoverage is defined as an extension of gml:RectifiedGridCoverageType (as an abstract XML schema type adv:AD_GitterCoverage).

A speciality of this mapping is, that these abstract classes of the base scheme do not directly extend gml:AbstractFeatureType but the mentioned gml-types. For this mapping a control parameter (“ExtendsType”) is introduced for the automated derivation of NAS (see below).

Instead of encoding the special coverage-properties of AA_PMO the corresponding properties gml:domainSet and gml:rangeSet of the mentioned predefined gml-types are encoded.

In the gml-profile which is used for NAS some more constraints are defined, e.g. for *rangeSet* only *file* and *dataBlock* are allowed.

The properties which are inherited from AA_PMO are mapped to predefined properties of the GML-types (gml:name, gml:description, gml:boundedBy).

The properties which are inherited from AA_Objekt are automatically included by the above described *copy-down* mechanism.

Encoding coordinate reference systems in NAS

In principle, each geometric unit must refer to a coordinate reference system (CRS, see 3.3.4.5) in the NAS file (point, line, surface) (s. 3.3.4.5). This can either be accomplisched by giving this reference on a higher level entity or by giving the reference on the concrete geometry entity itself. The reference is created by indicating a URI *(Uniform Resource Identifier)*. In order to avoid having to indicate this URI for each and every feature geometry, the used reference systems, one of which can be identified as a standard reference system, is stated in the Exchange Metadata of NAS. No further data on the coordinate reference system need be indicated for geometries existing within this standard reference system. In these cases, the "srsName" attribute proposed for GML-geometries is not present. The attribute must be assigned for all geometries not present in the standard reference system. The syntax described and identifiers defined in the document on Koordinatenreferenzsysteme für AFIS-ALKIS-ATKIS must be used.

Furthermore, the declaration of the coordinate reference systems used in the exchange metadata indicates the coordinate resolution and/or the number of relevant decimal places applicable to the reference system. This can differ from reference system to reference system and does not guarantee the accuracy of the coordinates. In AFIS-ALKIS-ATKIS, the coordinate resolution for metric position coordinates is specified at 3 decimal places (mm). It is necessary to indicate the relevant number of decimal places, because both GML and also ISO 19107 *Spatial Schema* make no restriction or provide any options in this respect (data type: *decimal* or *double*). The following definitions are used in the NAS schema files:

```
<xs:complexType name="AX_KoordinatenreferenzsystemangabenType">
    <xs:sequence>
        <xs:element name="crs" type="xs:anyURI"/>
        <xs:element name="anzahlDerNachkommastellen" type="xs:integer"/>
        <xs:element name="standard" type="xs:boolean"/>
    </xs:sequence>
</xs:complexType>
```

Selection options for representation

In principle, specification of "NAS Encoding Rules" is subject to a certain degree of encoding freedom; in view of unique interface description, however, this is utilised in just one specific way. All selection options within "NAS Encoding Rules" are formally described so that the XML schemas of NAS can be automatically and uniquely generated. This corresponds to the function of the Configuration Table in the "*XML Encoding Rules*" (ISO 19118 A.5.7).

The following requirement for information on additional regulations for controlling automatic generation is sufficient for automated generation of NAS:

- An object class to be exported in accordance with the standard regulations is not to be defined in NAS. Example: AC-classes.
- An abstract object class not to be exported in accordance with the standard regulations is to be defined in NAS. Examples: AA_NREO, AU_PunktObjekt.
- An attribute class to be exported in accordance with the standard regulations is not to be defined in NAS. Example: "identifier" in AA_Objekt, because in NAS, the XML-schema’s own mechanism in used.
- A role to be exported in accordance with the standard regulations is not to be defined in NAS. Example: Superfluous roles inherited from *GM_Object.*
- An attribute should use a pre-defined element. Example: "uposition" in the AU_pointobject is represented on element "gml:pointProperty".
- A role should use a pre-defined element. Example: "generator" in *TS_CurveComponent* is represented on element "gml:curveProperty".
- An attribute or role is to be represented by a special, local element. Example: Role "theme" in *TS_Feature* is represented via an element called "theme" and type "xs:string".
- In a complex case (geometric properties for AU_geometry object), the direct representation is nearby. In this case, attribute "uposition" should be linked as a

local *choice* element of the permitted "*properties*" ("gml:pointProperty", "gml:curveProperty", "gml:multiCurveProperty" and "gml:polyhedralSurfaceProperty").

- A feature type shall extend a predefined GML-*feature type* instead of gml:AbstractFeatureType. Example: Coverages.

The control information are also held in the application schematic in Rational Rose, i.e. for classes, attributes and roles, so-called "*Tools*" (file „nas.pty“) are defined, in which the control information are stored. The control parameters are recorded in file NASControlParameterReport.html.

The benefit of the solution based on *Rational Rose* is that all information is held in the application schema itself (and, may for example, also be reflected in the feature catalogue etc.).

The following control parameters are defined:

Table: Summary of control parameters defined for classes

Name	**Meaning**
Suppress	Indicates whether the class should be suppressed for the NAS definition, even if it had been defined in accordance with standard NAS Encoding Rules.
Consider abstract type	Indicates whether the class for the NAS definition should be considered, even if it had been defined as abstract classes in accordance with standard NAS Encoding Rules.
Consider ISOType	This control parameter is considered only for classes in one of the ISO-19100 packages and not for classes in AFIS-ALKIS-ATKIS packages. If the parameter is TRUE, the class (as well as all dependent classes) are incorporated into ISO19100.xsd.
Metadata	This control parameter is set only for classes in one of the ISO-19100 packages and not for classes in AFIS-ALKIS-ATKIS packages. If the parameter is TRUE, the class is considered in ISO19100.xsd.
ImportSchema	This control parameter is considered only for classes with stereotype "Request" or "Response". If the parameter is set, this line is inserted before the first element definition. It must be an <xs:import .../>-element.
Explanation	Optional explanation for the affected details so that the decision can be understood by thrid parties.
Extends Type	This parameter is used for complexTypes which shall extend a predefined XML-schema-Type by the xsd:extension-mechanism.

Table: Summary of defined control parameters for attributes

Name	**Meaning**
Suppress	Indicates whether the attribute should be suppressed for the NAS definition, even if it had been converted in accordance with standard

	NAS Encoding Rules.
Metadata	This control parameter is set only for attributes in one of the ISO-19100 packages and not in AFIS-ALKIS-ATKIS packages. If the parameter is TRUE, the attribute is considered in ISO19100.xsd.
Replacing element	If this parameter is indicated, the standard regulations are invalidated and the element defined in this parameter is used. It must be a <xs:element .../>-Element.
Referenced element	If this parameter is indicated, the standard regulations are invalidated and reference is made to the element defined in this parameter. It must be an element name and will in some cases have a name space declaration.
Explanation	Optional explanation for the affected details so that the decision can be understood by thrid parties.

Table: Summary of defined control parameters for roles in relations

Name	**Meaning**
Suppress	Indicates whether the role should be suppressed for the NAS definition, even if it had been converted in accordance with standard NAS Encoding Rules.
Metadata	This control parameter is set only for roles in one of the ISO-19100 packages and not in AFIS-ALKIS-ATKIS packages. If the parameter is TRUE, the role is considered in ISO19100.xsd.
ReplacingElement	If this parameter is indicated, the standard regulations are invalidted and the element defined in this parameter is used. It must be a <xs:element .../>-Element.
Referenced element	If this parameter is indicated, the standard regulations are invalidated and reference is made to the element defined in this parameter. It must be an element name and will in some cases have a name space declaration.
Explanation	Optional explanation for the affected details so that the decision can be understood by third parties.

10.1.3.5 Representation regulations for instances

This Chapter describes the representation of the instance model in corresponding XML elements. The result of the representation is a valid XML document (NAS-document).

Zipped XML-documents are also valid NAS-documents. The following zip-tools are allowed: “zip” and “gzip”.

The file contains:

- The *XML-header*, which is permanent: "<?xml version="1.0" encoding="UTF-8" ?>". The use of "UTF-8" is specified for the encoding.
- The *Root Element* from a request or response XSD file with a reference to the AdV-namespace "http://www.adv-online.de/namespaces/adv/gid/*version*" and the XSD-file.
- Elements in conformity with the referenced XSD file.

Each object in the instance schema is transferred to a corresponding element. The matching element bears the same name as the class to which the object belongs. The "gml:id"-attribute that carries the object identifier is used.

Each property of the object, i.e. each attribute and each role within an association, is represented in accordance with the representation defined in the schema regulations on XML elements, usually in a local element bearing the name of the attribute or role.

10.1.4 XML-schemas for de-jure standard sections and the external basic schema

The references to the complete XML schema files for the sections required from standards ISO 19100 and the basic schema are indicated below. The two files are available as separate documents:

ISO19100.xsd

AAA-Basisschema.xsd

10.1.5 XML-schema for AFIS-ALKIS-ATKIS

The XML-schema for the AFIS-ALKIS-ATKIS thematic schema is located in the following file:

AAA-Fachschema.xsd

10.1.6 GML-Profile for NAS

As a part of the NAS a GML-profile is defined and documented (gml3nas.xsd), which restricts the GML-elements and –types to the necessary scope and omits the parts of GML, which are not used in the actual version (like topology or unused object properties).

Beside omitting not needed GML-stuctures, some more decisions on the use of GML in NAS were made to restrict freedom on the kind of encoding and to make the use of NAS easier:

1. For GML-objects which allow an array-encoding beside the encoding of “normal” properties, the array-variant was deleted.
2. A gml:Curve which exists only of gml:LineStringSegments is equivalent to a gml:LineString. Not all GM_Curves can be represented by gml:LineString (e.g. if arc, circles or splines are used), but for the majority of geometries the possibility of

an encoding as gml:LineString shall be maintained. Out of that reason, both geometry types remain in the NAS. gml:LineString must be used if the GM_Curve has linear interpolation in all segments, gml:Curve must be used if there are segments which are of non-linear interpolation.

3. Analogously all surfaces in NAS are encoded as gml:Polygon (all surfaces have liniear interpolation); gml:Surface is not in the scope of the GML-profile of NAS.

4. Because the majority of area-objects (e.g. buildings, parcels), which are defined as GM_MultiSurface consist only of one area, analogously to 2. gml:_Surface has to be used if there is only one area and gml:MultiSurface only if there are two or more areas.

5. For the encoding of GM_Ring gml:LinearRing must be used if the ring consists of only one line-geometry with linear interpolation. In all other cases gml:Ring must be used.

6. With view to GML 3.1 gml:coordinates was omitted in the NAS-profile and the use of gml:pos provided. In GML 3.1 gml:coordinates will be replaced by a new element (gml:posList) and gml:coordinates will be a deprecated element.

7. For the time being gml:metaDataProperty is not used in the GML application schema to encode the object-related metadata. Out of that reason this property was omitted in the actual version of NAS.

10.2 NAS operations

NAS is in principle firstly designed for "outside", communication, i.e. for users of AAA-data. Depending on the implementation concept, it can also be used for the internal communication between recording and/or qualification systems and management systems. The last-stated functionalities are also considered in the following Chapters. An implementation that enables internal communication with system-specific functions, has to provide only those NAS operations from the range described that are relevant to the output of data to third parties. This includes specifically output of utilisation data and the management of a secondary database. As part of realising a network-based geodata infrastructure, it may also be necessary to provide further functions as NAS operations.

10.2.1 Scope of functions

NAS is to support various operations. The following requirements are currently identified:

- Set-up and updating of primary database
- Request for outputs
  - Output of utilisation data (extracts)
  - Management of secondary database (initial tagging and updating)
- locking and unlocking of objects
- Reservation (of point numbers etc.)
- Transfer of protocol information
  (e.g. processing protocols, error protocols)

Each NAS operation comprises two XML-schema definitions, one for requesting the operation and one for the response:

1. *Request*
   e.g. updating request, utilisation request
2. *Response*
   e.g. updating protocol, utilisation response

The multiple use of an XML schema definition for several operations is perfectly possible. If standardised XML schemas exist for the stated purposes, these are used, otherwise the definitions themselves are generated.

Like all other contents of NAS, the XML-schema definitions for NAS operations areautomatically derived from UML-models. For the AFIS- and ALKIS-applications, the UML models had already been generated (see Sections 6.2 and 7.4). Should it transpire that the definitions created here are also to be used for other applications, they should be incorporated at this location.

All XML-schemata for the NAS-operation are comprised in the file NAS-Operationen.xsd.

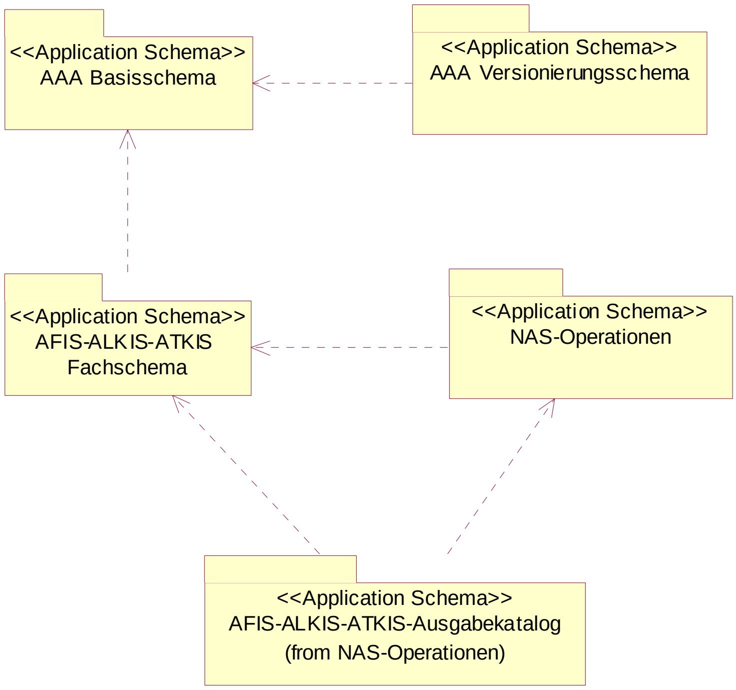


Figure 10-6: UML-package "NAS-operations" in the context of application schema components

10.2.2 Encoding the selection criteria

To code the selection criteria, in NAS, the `<wfs:Query>`-element from the specification *"Web Feature Service, Version 1.0.0"* is used in connection with the definitions in specification *"Filter Encoding, Version 1.0.0"* of the OpenGIS Consortiums.

The requirements for the selection and/or filter functionality of AFIS-ALKIS-ATKIS are more exacting than the functionalities currently described in the aforementioned specifications. In addition, therefore, the expansions explained below, which are currently **not** part of the aforementioned specifications in this form, are established[1]. The technical requirements have not yet been finally examined; this collection may necessitate further additions. The following expansions are currently defined for NAS:

- In GML, associations can be expressed by default either by hatching the referenced objects or via "`Xlink:href`" references. Both representations are in principle completely semantically equivalent[2] :

[1] See also Change Request (OGC-document 03-106). At the moment the document is only available for OGC-members.

[2] To make the NAS files easier to interpret, use of the 2nd representation in NAS is explicitly specified.

Representation 1:

```
<AX_Flurstueck>
   <istGebucht>
      <AX_Buchungsstelle gml:id="DEXXXX00000001">
         <zu>
            <AX_Buchungsstelle gml:id="DEXXXX00000002">
               <laufendeNummer>1</laufendeNummer>
            </AX_Buchungsstelle>
         </zu>
      </AX_Buchungsstelle>
   </istGebucht>
</AX_Flurstueck>
```

Representation 2:

```
<AX_Flurstueck>
   <istGebucht xlink:href="#DEXXXX00000001"/>
</AX_Flurstueck>
<AX_Buchungsstelle gml:id="DEXXXX00000001">
   <zu xlink:href="#DEXXXX00000001"/>
</AX_Buchungsstelle>
<AX_Buchungsstelle gml:id="DEXXXX00000002">
   <laufendeNummer>1</laufendeNummer>
</AX_Buchungsstelle>
```

For the first representation, an explicit pursuit of object associations is already explicitly permitted by the “/“ operator of *Xpath* in a Web Feature Service. Because these representations are semantically equivalent, it is explicitly permitted for NAS to have “/“ operator also affect `Xlink:href` references, where for the time being, only locally resolvable `Xlink:href`-references have to be supported. This means, for example, that an enquiry about the land parcel whose posting location is linked to another posting location numbered “1” via the “to” relation, can be formulated as follows:

```
<wfs:Query typeName="AX_Flurstueck">
      <ogc:Filter>
         <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>
               istGebucht/AX_Buchungsstelle/zu/AX_Buchungsstelle/laufendeNummer
            </ogc:PropertyName>
            <ogc:Literal>1</ogc:Literal>
         </ogc:PropertyIsEqualTo>
      </ogc:Filter>
   </wfs:Query>
```

- A recognised problem with Filter Encoding is the lack of appropriate support for ratings on multiple properties. Example:

```
<wfs:Query typeName="AX_Gebaeude">
   <ogc:Filter>
      <ogc:PropertyIsEqualTo>
         <ogc:PropertyName>weitereGebaeudefunktion</ogc:PropertyName>
         <ogc:Literal>1100</ogc:Literal>
      </ogc:PropertyIsEqualTo>
   </ogc:Filter>
</wfs:Query>
```

In this case, it is unclear which buildings are (to be) found: For example, do all other building function attributes have this value or must at least one attribute be set?

For the application in NAS, it has been explicitly agreed for the time being that appropriate ratings within the sense of “at least one attribute value satisfies the condition” are to be used.

- Use of `<wfs:XlinkPropertyName>`[3] . The WFS-basic.xsd is supplemented by the following definitions:

```
<xsd:complexType name="XlinkPropertyNameType">
   <xsd:complexContent mixed="true">
      <xsd:extension base="ogc:PropertyNameType">
         <xsd:attribute name="traverseXlinkDepth" type="xsd:string" use="required">
            <xsd:annotation>
               <xsd:documentation>
      Dieses Attribut gibt an, in welcher Tiefe Xlink:href-Verweise verfolgt und
      aufgelöst werden sollen. Ein Wert von "1" führt dazu, dass ein href-Verweis
      (auf ein lokales Objekt, bei entfernt liegenden Objekten muss die Auflösung
      nicht unterstützt werden) aufgelöst wird und das Zielobjekt mit in der
      Ergebnismenge zurückgeliefert wird. Hhref-Verweise aus diesem Zielobjekt
      werden wiederum nicht aufgelöst, da diese der Tiefe 2 zugerechnet werden.
      Ein Wert von "*" gibt an, dass alle (lokalen) href-Verweise aufgelöst werden
      sollen. Die erlaubten Werte sind positive, ganze Zahlen sowie der "*".
      Auch wenn ein Objekt durch mehrfache Verweise mehrfach aufgelöst wird,
      ist es nur ein einziges Mal in der Ergebnismenge repräsentiert.
               </xsd:documentation>
            </xsd:annotation>
         </xsd:attribute>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
<xsd:element name="XlinkPropertyName" type="wfs:XlinkPropertyNameType"
   substitutionGroup="ogc:PropertyName">
   <xsd:annotation>
      <xsd:documentation>
      Dieses Element darf in der NAS ein ogc:PropertyName Element ersetzen.
      </xsd:documentation>
   </xsd:annotation>
</xsd:element>
```

Use is permitted within the queries in the utilisation request and in the user profiles. The following regulations are emphasised:

- If the application schema requires (as in the case of NAS) that the object associations are always linked via Xlink references rather than inline, an Xlink traversal results in the referenced object being contained in the response quantity (feature collection).
- In the case of NAS, the resolution of href references explicitly supports the URN identifiers of the AAA-model.
- href references will for the time being only be resolved for locally available resources. Support for remote-Xlink resolutions will be supplemented at a later time if required.

- Use of `<wfs:XlinkPropertyPath>`. The WFS-basic.xsd is supplemented by the following definitions:

```
<xsd:complexType name="XlinkPropertyPathType">
   <xsd:complexContent mixed="true">
      <xsd:extension base="ogc:PropertyNameType"/>
   </xsd:complexContent>
</xsd:complexType>
<xsd:element name="XlinkPropertyPath" type="wfs:XlinkPropertyPathType"
   substitutionGroup="ogc:PropertyName">
   <xsd:annotation>
      <xsd:documentation>
      Dieses Element darf in der NAS ein ogc:PropertyName Element ersetzen.
      Es erlaubt (im Unterschied zu wfs:XlinkPropertyName, das die generelle
      Expansion bis zu einer bestimmten Tiefe unterstützt) die gezielte Auflösung
      von Xlink:href-Verweisen entlang eines bestimmten „Eigenschaftspfades“.
```

[3] See OGC-Document 02-063, the document is currently only available to OGC members.

```
        Als Wert wird eine Xpath-Pfadangabe verwendet, bei der das Ende ein
        Objekt steht, bei der die Auflösung abbricht.
        Beispiel: Ein XlinkPropertyPath
          "istGebucht/AX_Buchungsstelle/istBestandteilVon/AX_Buchungsstelle"
        bei einer Query auf AX_Flurstueck führt dazu, dass die Buchungsstelle und das
        Grundbuchblatt in der Ergebnismenge für jedes selektierte Flurstück direkt
        mit zurückegeliefert werden.
        </xsd:documentation>
     </xsd:annotation>
  </xsd:element>
```

10.2.3 Set-up and updating of a primary database

Because GML itself offers no elements for updating operations, the definitions from the *Web Feature Service* (WFS) of OGC are used for this purpose. Besides the transaction mechanism, the WFS-specification defines 3 change functions: <insert> (insert new object), <update> (amend, overwrite object) and <delete> (delete object). In the feature catalogues is to specify which changes result in <update> or <delete> followed by <insert>. This applies to both changes in the attribute values and relations and also to geometric changes. In the case of the latter, the processor in the collection process may have to decide which updating type is to be used.

Example:

```
<wfs:Transaction>
  <wfs:Insert>
    <AX_Flurstueck gml:id="DEBY0000F0000001">
       ...
    </AX_Flurstueck>
    <AX_Gebaeude gml:id=”DEBY0000G0000001”>
       …
    </AX_Gebaeude>
  </wfs:Insert>

  <wfs:Update typeName=”AX_Flurstueck”>
          …
    <wfs:Property>
      <wfs:Name>amtlicheFlaeche</wfs:Name>
      <wfs:Value>287.3</wfs:Value>
    </wfs:Property>
       …
    <ogc:Filter>
      <ogc:FeatureId fid="DEBY0000F000000220010101T000000Z"/>
    </ogc:Filter>
  </wfs:Update>

  <wfs:Delete typeName=“AX_Buchungsstelle”>
    <ogc:Filter>
      <ogc:FeatureId fid="DEBY0000B000000320010101T000000Z"/>
      <ogc:FeatureId fid="DEBY0000B000000420010101T000000Z"/>
    </ogc:Filter>
  </wfs:Delete>
</wfs:Transaction>
```

The following has to be noted:

- The filter expressions for <delete> and <update> operations may contain only *FeatureId* elements. More complex filter criteria are not permitted.
- In *FeatureId* conditions in "<update>" and "<delete>" operations, the identifier must be supplemented with creation date and time, so that the version and the actuality can be tested and the correct version of an object can be addressed. In this place date and time are encoded without separation characters to fulfil the conditions of a gml:id, i.e. in the form: CCYYMMDDThhmmssZ. For this rule there is the following exception: Within an update request with more than one <update>-or <delete>-case (*"Fortführungsfall")* for features which are multiply updated the expansion of the oid with date and time can be omitted from the 2nd update case on. Only in this special case there is no testing on version and actuality, but the actual –just created- version is used.In <delete>-operations it is only possible to handle more than one feature if all those features belong to the same feature class. In <insert>-operations it is possible to handle features of more than one feature class. <insert>-operations always handle only one feature.
- <update>-operations always include all properties of an AAA-feature, not only the changed ones. This is a restriction on the WFS-specification, which demands that at least all the changed properties are included. The reason for this restriction is that it is difficult for database systems to track changes of single properties, and that it is much more simple just to track changes of features as a whole.All changes carried out within a updating are valid at the same time. The system time (transformed to UTC) at the start of the transaction is entered into the "lifetime interval" of the object attribute. The start or the end date is to be assigned on a case-by-case basis. Any data supplied for the individual thematic objects are irrelevant and will be overwritten. The latter does not apply for the initial set-up of a database if objects are transferred from a previous database (migration). In this case, the date provided can be used. If the date/time "9999-01-01T00:00:00Z" ("dummy date") is provided it is overwritten with the system time. All time-information is provided in UTC time (Universal Time Coordinated, Greewich Mean Time). The time unit for entries in the lifetime interval (data type: *DateTime*) is a full second and including the mandatory character "Z" for UTC (CCYY-MM-DDThh:mm:ssZ). The management system ensures at transfer that 2 versions of the same object with identical lifetime intervals cannot arise. This may occur if an object is changed within a updating request into several updating cases and due to the system speed, these are processed during the same second.

The XML-schema for a updating request and its response are, like all other NAS-operations, contained in the file NAS-Operationen.xsd. Set-up requests and their responses are sub-classes of updating requests. The installation and updating of metadata datasets is not yet included in this modelling. It will be added as soon as the requirements are defined.

The functions for updating are run differently in systems with a full historical database and in systems without complete history:

Systems without complete historical database

<insert>:
The transferred thematic objects are entered as new information.

<update>:
The transferred thematic objects replace the thematic objects that have the same identifiers. For unique identification of the version to be overwritten and/or versioned, the identifier (XML-attribute *fid*) of the new thematic object in the filter expression is supplemented by the creation date/time data for the object version to be overwritten. This should reveal any errors occurring through updating requests that do not match the stored database. The thematic object is stored again with the original (not supplemented by creation date/time) identifier in the receiving system. It is not permitted to replace the operation *<update>* with *<delete>* and then *<insert>* with the same identifier.

<delete>:
The attribute *fid* of the filter expression in the WFS<delete> element identifies the thematic object to be deleted. To uniquely identify the version to be deleted, the identifier is supplemented in the exchange file by the creation date/time of the version to be replaced. This should reveal any errors occurring through updating requests that do not match the stored database. The object thus described is deleted from the receiving system with all self-referenced properties and referenced spatially-referenced basic forms. Spatially-referenced basic forms are deleted only if they are not referenced by any other object.

This functionality is used by all data-management systems that hold secondary data inventories.

Systems with a complete historical database

If the receiving system is configured to manage a complete historical database, it responds to:

<insert>
by producing a new instance of a container for feature versions and inserts an initial version of the transferred thematic object into the container.

<update>
The transferred thematic objects are entered into the container for feature versions identified by the identifier as a new version. For a unique designation of the previous version, the identifier in the filter expression (XML-attribute *fid*) of the new thematic object in the exchange file is supplemented by the creation date/time data for the object version to be overwritten. This should reveal any errors occurring through updating requests that do not match the stored database. The overwritten thematic object remains in the receiving system as a historic version.

<delete>
The version of the thematic object identified by the identifier expanded by the creation date/time (XML-attribute *fid*) in the filter expression is tagged with the current expiry date/time (derived from the system time) and historized. The system ensures that no other versions can be created.

This functionality is also used by data management systems that use temporary versioning for the provision of updating information for third parties as part of the NBA procedure (see below).

The conceptual thematic model for updating of ALKIS and the exact procedures for their updating processing (provisional) are described in Section 5.3.

10.2.4 Request for outputs

The data to be output by a data management system (utilisation data or data for managing a secondary database) are determined in respect of the scope of information to be output by selection and filtering criteria. A data management system must therefore be able to evaluate complex selection and filter expressions and output the data to be qualified by them. **Selection** takes place through spatial, technical (feature type, attribute, relation) and temporal criteria. These criteria can also be boxed and connected together so as to create entire selection chains. Which elements follow other elements through references for output can then be formulated.

Filtering criteria determine which elements of the selection chain are to be output and which attributes and references are output with these elements.

The selection and filter criteria are transferred as a component of the utilisation requirement for the data-managing system and logged there in the user profiles. Standardised selection and filter criteria are defined for the definition of standard AdV products (e.g. in Chapter 7.2).

As the formal language for defining the selection chains the *Filter Encoding Specification* of OGC is used.

The XML-schema for a utilisation request is contained in the file NAS-Operationen.xsd. It uses the schema filter.xsd of OGC.

10.2.5 Output of utilisation data

The output of utilisation data is a data output without explicit indication of a functionality to be performed within the receiving system. A special preparation of data in dependence of the output requirement (e.g. production of "land parcel-centred view" in ALKIS) is possible by appropriate temporary objects being output (see Section 7.2).

For the utilisation response, the FeatureCollection (WFS-basic.xsd) from the *Web Feature Service* of OGC is used and accordingly, supplemented for AAA by further information. Depending on the utilisation request, an internal schema file is used for each type of output.

10.2.6 Managing a secondary database

A secondary database is managed via the user-related update of primary database (Section 10.5) based on updating case (continual) or effective date (by grouping several updatings on the same object into one single updating). The operations *<insert>*, *<update>* and *<delete>* are practically performed by the receiving system in the same way as for the management of primary database.

The statements made on outputs (Item 10.2.4) also apply for the schema file.

10.2.7 Locking and unlocking of objects

Locking requests enable objects in the management to be locked against updating by third parties by stating a list of object identifiers. Unlocking requests clear the lock.

10.2.8 Reservations

To reserve codes (e.g. for surveying points, land parcel identification etc.), appropriate requests can be formulated for a management system. The response file provides a list containing the requested codes.

10.2.9 Transferring protocol information

Because for each NAS operation, both a *Request* and a *Response* class are defined, the latter defines which protocol information is output for the respective operation. They are therefore contained in the individual operations.

10.3 Units to be exchanged

The smallest units for data exchange are complete thematic objects. This also applies in principle for the updating of the primary database (AAA management system). Irrespective of whether objects have qualified for output by their own properties or by the evaluation of a defined selection chain, in terms of updating functionality, they should in principle be regarded as independent updating units (for exceptions, see Section 10.4).

Utilisations that do not serve the purpose of updating the primary database, may depending on user requirement or user profile, create incomplete thematic objects (missing attributes or relations) or "temporary objects" created through special data preparation for data exchange.

Data is exchanged in NAS, independent of the conceptual model used for versioning (container with versions), as if all object versions were independent objects. It is thus possible to identify in an identical manner the data exchange interface for locations that hold a complete history and those that do not. The following framework conditions, however, must be observed:

- To reduce the number of created versions, reciprocal relations in the data exchange must be presented by a single, unilateral relation. The relations in the data exchange that are coded are those that have been defined in the UML schema as a preferred navigation direction. Reciprocal relations in the standardised schemas are replaced by suitable unilateral configuration.
- So that during data exchange, the version of an object to be overwritten and/or versioned can be uniquely identified, the identifier in the exchange file is supplemented in the XML-<delete> and -<update> elements by creation date and time. It is only necessary to supplement the identifier by the time stamp during a data exchange to ensure that updatings also relate to the current database. In the database itself, the versions to be referenced are obtained by evaluating the lifetime interval of the versions at attributive level.

10.4 Implicit functionality

The updating of the primary and secondary database via the NAS interface requires that the receiving system, besides execution of the explicit functions <*insert*>, <*delete*> and <*update*> (see 10.2.1.1) also has implicit functions, which permit convenient operation with the system.

The scope of the implicit functionality of a data management system to be realised varies between systems for primary and secondary databases. The number of functions to be requested from the data user by a secondary database system should be as low as possible, to enable simple implementation. Conversely, a data management system for the primary database may have significantly more functions at the originally responsible data-managing location.

10.4.1 Implicit functionality of a system for the primary database

When the NAS is used to communicate between a qualification and/or recording system and a management system, the following implicit functions are required:

- The receiving system derives on entering the new versions the **creation date/time** from the system time. All new versions entered for a updating (or created by the <*update*> function) have the same creation date/time. This is usually the time at which the transaction is started (*commit).* If a request consists of part requests (updating cases), they are processed in the sequence in which they appear in the NAS file. Each part request is assigned its own creation date/time.
- During data exchange via NAS, references are exchanged only unilaterally in the preferred direction of the reference. The receiving system implicitly creates the **counter-reference.** No new version is formed when the counter-reference is created.
- There are thematic objects that only have a right to exist if they are referenced by other objects (e.g. position-type objects). Because counter-references are not transferred via NAS, a updating system cannot know whether an object that is no longer referenced by the updating can also be deleted. The **thematic objected no longer being referenced** must be **deleted** by the database. The thematic objects that can be deleted due to a lack of referencing must be identified in the feature catalogue. This type of updating is used for versioning and archiving.
- There are thematic objects that reference objects to be deleted as part of the updating process. Because counter-references are not transferred via NAS, a updating system cannot know whether an object to be deleted is referenced by further objects. As a result, it may occur that references are no longer satisfied

following updating. The data management system must **automatically delete such unsatisfied references.** This type of updating is used for versioning and archiving.

- There are thematic objects that only have a right to exist if they reference other thematic objects (e.g. presentation objects). If as part of a updating all such references are explicitly or implicitly deleted, the data management system automatically **deletes** the corresponding **thematic object that lacks the necessary references.** The thematic objects that must be deleted due to a lack of the necessary referencing must be identified in the feature catalogue. This type of updating is used for versioning and archiving.
- If only the technically changed objects are provided during a updating (e.g. only the deleted and the new land parcels when splitting up a land parcel), the database creates the topological consistency by breaking-up and/or separating the geometries of the objects affected only indirectly (“**implicit geometry handling**”, see section 10.4.1.1).
- When **deleting geometries,** break-ups from previous implicit processes are to be reversed again according to the following regulations. A position is removed from the geometry of all objects, if it contributes to no object in which it is used for the geometric definition of this object; if it contributes to the geometric definition of one object only, it is retained in all objects. A position contributes to the geometric definition of an object if the object has a point spatial reference or if (with line or surface spatial reference) it does not lie in a straight line with the previous and following position. The term "lies in a straight line" should be defined in dependence of the specified co-ordinate resolution (for metric position co-ordinates in AFIS-ALKIS-ATKIS: millimetre). In the accepting system, this implicit behaviour results in updatings that are not explicitly indicated in the updating request to be triggered from NAS. These updatings are ~~not~~ implicitly to be triggered by the receiving system and result in new versions of all participating objects being created.
- If for the updating of a primary database, exchange elements with provisional identifiers (see 3.3.7) are supplied, the receiving system **generates** final **unique identifiers.**
- At locations that do not hold a complete history, the data manager automatically **generates** the associated "Historical land parcel" object when a current land parcel is deleted.
- Further implicit functions (e.g. assignment of point characters) are **implementation-specific.**

10.4.1.1 Implicit Geometry Handling

"Geometry handling" is an implicit functionality of the AAA-database which is used for updating processes. In this context new or changed geometries are connected with existing geometries in this way that depending on the defined themes identical (redundance free) are produced.

This functionality is necessary if the editing software (*"AAA-Verarbeitungskomponente")* does not transfer all (indirectly) changed features to the database (e.g. only the deleted and the new land parcels when splitting up a land parcel). For the geometry handling the following rules are agreed:

- The functionality of geomtry handling can be optionally realized by the AAA-database systems (*"AAA-Führungskomponente").* The editing systems (*"AAA-Verarbeitungskomponente")* can then make use of this functionality. If an AAA-database system decides not to realize the geometry handling functionality, all editing systems that communicate with that database have to deliver complete date (i.e. all changed features). Not affected by this is the commitment of the database to test the geometric consistency of the data.
- All features which are only implicitly changed by geometry handling in the database are versioned.
- The geometry handling can only be applied to features within class themes: a geometry handling for features within instance themes is not intended. For this reason a geometry handling within class themes has no impact on instance themes (no "cascading" geometry handling).
- For features within instance themes the editing system (*"AAA-Verarbeitungskomponente")* has to take care of the following points: a) if a geometric identity is wanted, lines have eventually to be splitted b) all changed features have to be included in the update request, including the only indirectly changed ones.
- AX_Fortfuehrungsauftrag ("update request") includes a steering parameter (geometry handling yes/no). The AAA-database has to evalute this parameter and turn on or off the geometry handling functionality. The editing system has to guarantee that the parameter value is identical to the content of the update request.
- A geometry handling for receiving systems in the NBA-procedure is not expected/required. All changed features are transferred to the secondary database, including the indirectly changed ones.

The following geometric criteria are valid:

- The search radius for the geometry handling is square root of 2 [mm]
- All points, vertices and lines take part in the geometry handling.
- For lines, only strait lines and arc/circles take part in the geomtry handling. Spline-geometries do not take part in the geometry handling; for these geometries the editing system has to take care that all affected features are marked for updating.
- In the case that a new line is introduced, an existing point which lies “under” this line has to split up the line. The line must include the point.

10.4.2 Implicit functionality of a system for a secondary database

The management of a secondary database (see Section 10.5) via the NAS interface involves the receiving system (if required by the user) creating and maintaining the counter-references to the exchanged references.

Update commands for which the object to be revised is not yet in the user’s database are to be treated as *insert* commands on transfer. (Example: A user receives all land parcels and the associated owners in the respective area of interest. A land parcel changes its owner. From the user’s point of view, the owner is new (*insert)* but from the point of view of ALKIS management system old (*update),* because he already owned land parcels outside the area of interest and has therefore been managed in the outputting system for some time although never managed in the user’s system).

10.5 User-specific updating of secondary database (NBA)

10.5.1 Technical requirements

The technical requirements of the user-specific updating of secondary database (NBA) are based on the procedures that exist in ALK/ATKIS- and in the ALB-system.

Technical requirements can be summarised as follows:

Change data are to be derived on the basis of revision data, which show the structure of the primary database data. Change data for user-specific updating of secondary database

1. be capable of continuous and revision-related (change data) and/or
2. effective date-related (change-only data) output.

Revision-related means that all changes which have taken place in a previous period are indicated according to the time sequence. This makes it possible to transfer all processes step-by-step into the receiving system. The pre-condition, however, is that all information concerning inserting, updating and deleting objects in the previous period is contained in the change data.

Conversely, the effective date-related procedure provides only the change-only data required to bring the starting status of the user to the actual status required by the user. In this case, it is not necessary to transfer what happened to the objects before the actual status is reached. The effective-date related change-only data represent a sub-quantity of the change data and may only be derived from them through evaluation; they comprise all newly created objects, the latest versions of revised objects and details on historized objects.

For each user, a profile is created that describes the criteria according to which the user is to be provided with change data from the database once held for the NBA procedure. User-related selection criteria are:

1. Technical through the indication of feature types, attribute types and values and relations
2. Spatial through the indication of an area and
3. Temporal through the indication of a time interval.

Feature types, attributes and relations also define content of the data delivered to the single users; these details are also to be logged in the user profile, which is, realised by feature type AX_user group.

10.5.2 Modelling

The NBA procedure is to be offered for all feature types that a data-provider keeps in in his database. The user-specific selection can be based on the whole scope of object properties; the data protection requirements has to be considered. As a result, the NBA procedure always delivers objects as the smallest units of change data. These data refer entirely to the current user profile; as far as the entire database is concerned, these objects may also be incomplete. If revision data for the same time interval are transfered in several portions, the outputting system ensures that samethe version of a technical object is delivered to the user only once.

The spatial expansion of the user's area of interest is described by areas in the user profile. Spatially-related elementary objects (REO) are qualified as soon as a part of them lies within the requested area. The degree to which objects can be followed by tracking relations must also be described in the selection criteria of the user profile.

The time period for which the provision of change data has to be guaranteed for various users according to the NBA procedure can be limited (temporal framework). This makes it possible,

1. for each user to retrospectively request change data within this time period and
2. to provide change data on a user-related basis without having to store them user-related data.

The procedure for the user-specific update of the secondary database requires information on database changes to be held for this period. The period is determined by the data-provider in agreement with the users.

The management of an object's various characteristics over time required for the NBA procedure is realized by the version concept. Therefore,

- change data is managed at primary database level,
- the management of information for the procedure of an user-specific update of the secondary database is based on the version concept and
- no new, additional and therefore redundant data structure is developed.

This makes it possible,

- from a collection of changes,
- which has to contain complete information on the database objects,
- over a period of several years (in dependence of the temporal framework)
- to conduct evaluations according to
  - scope of content through feature types, attributes and relations,
  - spatial expansion through areas and
  - temporal expansion through time intervals
- and

- on a user-related basis.

In order to identify the version to be overwritten and reveal transfer errors in the NBA procedure, it is necessary to supplement the object identifiers during data exchange by the creation date/time. This calls for the following requirements:

- The creation date in the object identifier can be omitted during implementation (e.g. in the receiving system) (replaced by time stamps of the versions)
- For data exchange via NBA procedure with revision-related (continuous) data output, the relation with a creation date for the referenced information that matches

the creation date of the object version is output during the exchange of object versions.

- For data exchange via NBA procedure with effective-date related (change-only data), the relation with a creation date for the referenced information that matches the effective date is output during the exchange of object versions.
- When the exchange file for the user-related update of the primary database is generated, the outputting system performs the following functions:
- Selection of the data to be output from the (if necessary temporary) historical database in accordance with the selection change and filter details logged in the user profile,
- Generation of the revision operations for the receiving system from the historical database,
- Conversion of the data in the norm-based data exchange interface.

For derivation of the revision operations to be created, the object qualifying for data output must be evaluated to confirm whether it is a first, further or final version from the point of view of data management.

10.5.2.1 Output of change data

During the continuous, revision-related data output (change data), all versions of an object qualifying for data output are processed. The time interval under consideration spans from the last data output to the present as a maximum. From a data management point of view, this must be evaluated to confirm whether it is a first, further or final version of an object.

Qualifying version from a primary database point of view	**Output operations**
first version of a new object	*<insert>*
further version of an object	*<update>* of last transferred version (indicate create date/time)
final version of an object	*<delete>* of last transferred version (indicate create date/time)

10.5.2.2 Output of change-only data

During effective-date related data output (change-only data), only the latest or last version is processed under the versions of an object, whose creation and/or expiry date lies within the time period under consideration.

Latest or last qualifying version from a primary database point of view	**Output operation**
first version of a new object	<*insert*> the current version of this object
Further version of an object	<*update*> of last transferred version (indicate create date/time) with the current version of this object
Last version of an object	<*delete*> of last transferred version (indicate create date/time)

11 Quality assurance

11.1 AdV quality assurance system

The AdV has agreed the following key points of the quality assurance system for the geodata of official surveying and mapping:

"Through national regulation, designation and descriptive, quantitative quality features, the AdV identifies and guarantees the quality of the geotopograpical and real-estate descriptive products of official surveying and mapping. National topicality, uniformity, completeness and availability of the products are essential characteristics in this regard. The surveying authorities guarantee compliance with AdV product quality by standardised test procedures and declare conformity with the AdV standards.

The objective is a comprehensive quality assurance for the geodata of official surveying and mapping as a result of the conception and production process. The conception (AAA-basic schema, AAA-technical schema) is task of the state communities represented by the AdV, during which production of the data inventories in harmony with the AAA-application schema is the task of the surveying authority of each individual state.

11.2 Quality assurance model

The relationship structure of the aspects to be quality tested is shown in the following quality assurance model for the AAA-application schema:

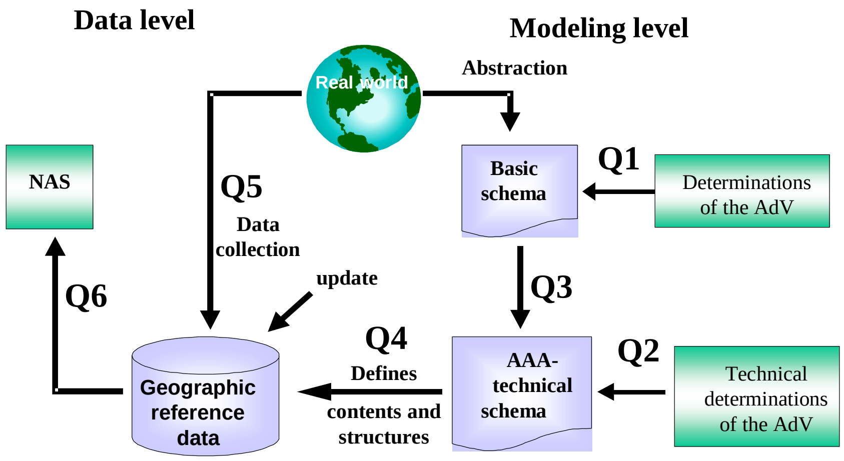


Figure 11-1: The quality assurance model of the AFIS-ALKIS-ATKIS project

Q1 measures the AAA basic schema against the strategic and technical stipulations of the AdV, Q2 measures the AAA technical schema against the technical stipulations of the

AdV. Q3 determines whether the AAA technical schema corresponds to the regulations of the AAA basic schema. Q1, Q2 and Q3 verify the conceptional, internal quality.

Q4 verifies the geobasis database internally as a product for logical agreement with the AAA application schema and compliance with the defined quality specifications, while Q5 compares the geodatabase externally with the real world. Q6 relates the quality of the NAS to the user.

The following quality testing schema is derived:

		AdV	States
1.	**AdV regulations and standards for the development of procedures and program systems**		
	Quality assurance of the AAA-basic schema against stipulations of the AdV (Q1)	X	
	Quality assurance of the common AAA-technical schema against the technical stipulations of the AdV (Q2)	X	
	Quality assurance of the common AAA-technical schema against the AAA-basic schema (Q3)	X	
	Quality assurance of data inventories (ALKIS/ATKIS/AFIS) against the common AAA-application schema (Q4)		X
	Quality assurance of the exchange data against NAS (Q6)	Fundamental principles	X
2.	**Stipulations for AdV product quality**		
	Stipulation of descriptive and evaluating quality features for unified products including topicality, uniformity, completeness and availability	X	
3.	**Stipulations for quality assurance of the primary database data**		
	Quality assurance of the primary database data against technical reality (Q5)		X
4.	**Quality assurance (as part of quality management)**		
	Conformity declaration by the surveying authority		X

The quality assurance principles for Q6 assume that when data is submitted from AFIS®/ALKIS®/ATKIS®, the created NAS files do not have to be checked against the model. The model-compliant implementation must guarantee this using the valid XML schema files (XSD); interoperability must be guaranteed. Data acceptance is part of the qualification process. For this purpose, appropriate test tools must be provided which ensure the required quality of the accepted data by using the currently applicable XML schema files (XSD). The testing of exchange data against the NAS-schema is differentiated between testing for good shape of the XML-file (test tool e.g. xmlint.exe) and testing for validity of the XML-file (test tool e.g. XSV.exe).

11.3 Systems and recording of quality assurance

On the basis of ISO 19105 “Geographic Information - Conformance and testing“, abstract test suites (ATS) are to be formulated and used to examine conformity. Each AAA-quality criteria can then be analysed and recorded according to the following schema:

- Theories (conformance requirements).
- Examination solutions, formulated as questions.
  Each of the questions can result in separate test modules and test cases, which are structured as follows:
  a) Test purpose
  b) Test method
  c) Reference
  d) Test type.
- Test for confirming or refuting these theories (executable test suite – ETS with executable test cases).

The results of quality assurance testing for the AAA-application schema are published in the following documents under http://www.adv-online.de/veroeffentlichungen:

- „Aufstellung von Qualitätskriterien und Aussagen zu deren Umsetzung für die GeoInfo-Dok“, Status 26.10.2000.
- „Grundsätze für Qualitätskriterien und standardisierte Prüfverfahren für die Anwendung des AFIS-ALKIS-ATKIS-Basisschemas bei der Entwicklung der Fachschemata“ including Appendix “Quality criteria (Q3) for examination of the AFIS-ALKIS-ATKIS technical schema against the AFIS-ALKIS-ATKIS basic schema“; Status:10.07.2002.
- Documentation on the Modelling of Geoinformation of Official Surveying and Mapping (GeoInfoDoc) - - Qualitätssicherung ALKIS -; Status 01.03.2003.
- The documentation on quality assurance for AFIS and ATKIS is not yet available.

12 Glossary, abbreviations

12.1 Technical terms and their English translations

Technical term (German)	Explanation	Technical term (English)
AdV-standard	The AdV creates instruments for the development of *Procedures* and *Program systems* and for the creation of *Products*. AdV instruments, which are used to specify standardised state core data, data exchange interfaces and standard products, are examined for their compliance with AdV-standards through the obligation of the member authorities.	**AdV Standard**
AFIS-ALKIS-ATKIS-Anwendungs-schema	The basic schema and the application-specific sub-schema of AFIS, ALKIS and ATKIS (AAA-thematic schema) together form the common AFIS-ALKIS-ATKIS application schema.	**AFIS-ALKIS-ATKIS application schema**
AFIS-ALKIS-ATKIS-Basisschema	→ see basic schema	**AFIS-ALKIS-ATKIS basic schema**
AFIS-ALKIS-ATKIS-Referenzmodell	The AFIS-ALKIS-ATKIS reference model is a common framework model, in which the structures and contents of the AFIS, ALKIS and ATKIS products, the data recording sources, primary database data and also their digital and analogue extracts from AFIS, ALKIS and ATKIS and also output of data to the users are defined as components with their reciprocal relations.	**AFIS-ALKIS-ATKIS reference model**
Anlass	The cause describes the reason for a change to an object. It appears as an attribute for AA-object next to the object identifier and the lifetime interval.	**cause (for a change)**
Anwendungsschema	An application schema is a conceptual schema for data required by one or more applications. conceptual schema for data required by one or more applications	**application schema**
Attribut	Attributes are self-referenced properties of an object. Their individual structures are described for each feature type as an attribute type in the feature type catalogues.	**attribute**
Ausgabekatalog	The output catalogue specifies the method of preparing and outputting the data and extracts from AFIS, ALKIS and ATKIS to the user.	**output catalogue**
Auszüge	Extracts are data inventories selected according to content, area and/or time period (e.g. updating data inventories), which are output to the user as data	**extracts**

	structured by object or image or as analogue extracts.	
Basisschema	The basic schema is a schema that describes the fundamental properties of one or more applications. It contains the unified and object-oriented model solution, on which the sub-schemas of AFIS, ALKIS and ATKIS are based.	**basic schema**
Bestandsdaten	Primary database data is geoinformation on official surveying and mapping in AFIS, ALKIS and ATKIS. They contain the complete description of thematic objects, including the data on their cartographic or textual depiction in one or several target scales.	**(geographic) data in primary database**
Bestandsdatenaktualisierung	The primary database update is a process for revising secondary databases for users using the norm-based data exchange interface (NAS). The process is abbreviated to "NBA - Process".	**update of primary database**
Bestandsobjekte	Primary database features are features of the real estate cadastre modelled according to the AFIS-ALKIS-ATKIS data model.	**features in primary database**
Datenmodell	A data model describes the fundamental properties, which make a unified representation for all appearances of a certain view of reality. It determines the fundamental structures, the potential relations and the properties that can be assigned. → see also model	**data model**
Daten-modellierungs-sprache	→ See conceptual scheme language	**data modeling language**
Differenzdaten	Change-only data are change data relating to the effective date that are required in order to bring the initial status of the primary database data up to the required final status (effective date) for the user. They comprise all newly created objects, the latest versions of revised objects and details on historized objects. The differential updates represent a sub-quantity of the change data.	**change-only data, differential update**
Digitales Bildmodell	A digital image model is a model for storing image data, e.g. digital orthophotos.	**digital image model**
Digitales Geländemodell	A digital terrain model is a digital elevation model with additional topographic information as break lines etc.	**digital terrain model**
Digitales Höhenmodell	A digital elevation model stores information about the elevation of distinct points, which, in most cases, form a rectangular grid. These elevations are used to compute/interpolate elevations for all other positions.	**digital elevation model**
Elementar-objekte	Elementary objects are the smallest, technically independent units. They are not composed of other indpendent units. The following types of elementary objects exist in modelling for AFIS, ALKIS and ATKIS: Spatially-related elementary objects (REO) Spatially-referenced elementary objects are to be	**elementary objects**

	formed if, in addition to technical properties, geometric or topological properties are to be demonstrated. Non-spatially related elementary objects (NREO) Non-spatially referenced elementary objects are to be formed if, in addition to technical properties, no geometric or topological properties can also be shown. → See also “Composed objects (ZUSO)”	
Erhebungsdaten	The collected data represent the basis for revising the official geoinformation. They are formed by collection from source data, which are collated through reconnaissance methods in the real world or recorded from cartographic representations and other documents.	**collected data**
Erhebungs-prozess	For qualification and updating of the official geoinformation, the data collection process generates collection data from source data. The collection process is not part of the ALKIS application schema and is modelled specifically for each state.	**data collection process**
Fachdaten	Technical data are application-specific data of a technical user, e.g. managing data or customer data of a utility company. These can be tagged with a spatial reference.	**technical data**
Fachdatenobjekt	Technical data objects are objects in technical information systems of other technical areas.	**technical data object**
Fachdaten-verbindung	The technical data connection comprises the integration and linking options between the data of the surveying authority (basisdata) and the technical data in the form of references. This linking can take place in the spatially-related basic information systems of the surveying authority or in the technical information system (unilateral linking) or reciprocally in both information systems (reciprocal linking).	**association to technical data**
Fach-informations-system	System containing as its basis information of a technical nature and geobasis information of the surveying and cadastre authority.	**technical information system**
Fachobjekt	A feature is created through abstraction of an object or fact in the real world. In the application area of AFIS, ALKIS and ATKIS, this is limited to the objects and facts that make up the technical content of AFIS, ALKIS and ATKIS. → Object abstraction of real world phenomena NOTE 1 A feature may occur as a type or an instance. Feature type or feature instance should be used when only one is meant. NOTE 2 UML uses feature for another concept than the use of feature within this standard. In UML, a property, such as operation or attribute, is encapsulated as part of a list within a classifier, such as an interface, a class or a data type.	**feature**
Festpunkt	A control station is a point on the ground whose position (horizontal, vertical) is used as a base for a	**geodetic control**

	dependant survey.	**station**
Fortführung	Updating means the updating of primary database data. The updating data (data and metadata) involved are transferred to the database by applying suitable methods.	**update, updating**
Fortführungs-auftrag	The updating case is a feature type composed in one or more updating cases to become one unit. It controls the process of data updating for all database objects.	**updating case or instance**
Führungsprozess	The updating process relates to the initial set-up and updating of primary database data (geobasis data and metadata).	**process of updating**
Geobasisdaten	Reference data are official geodata that describe the landscape (topology), sites and the buildings in the unified geodetic spatial reference independent of application. Geobasis data are examined and prepared by the state surveying authority. They fulfil the function of the basisdata for geotechnical data.	**(geographic) reference data**
Geodaten	Geographic data (geodata) are data that refer to spatial objects in relation to the earth's body.	**geographic data**
Geo-informationen	Geoinformation is geodata that is selected, edited and collated for a specific application.	**geoinformati on**
Geo-informations-system	A geoinformation system is a system that records, stores, checks, changes, integrates, analyses and represents geoinformation.	**geographic information system**
Geokodierung	Geocoding refers to the assignment of objects (data, information) to the earth's surface using a (spatial) reference system.	**geocoding**
Grunddaten-bestand	Core data refer to the database provided by all surveying authorities of the states of the Federal Republic of Germany (in AFIS, ALKIS and ATKIS) for all users throughout the country.	**(geographic) core data**
Historisierung	Historization describes the creation of the last version (expiry) of a thematic object.	**historization**
Identifikator	The identifier uniquely identifies an object (unique). It is a special self-referenced property of the object and acts on behalf of the object that it represents. It remains unchanged for as long as the corresponding object exists. The exchange interface defined for the AFIS-ALKIS-ATKIS data exchange is based on the application of ISO 19118 Encoding. The data exchange interface that is therefore norm-based is abbreviated to "NAS".	**identifier**
Kardinalität	Cardinality is the power of a quantity and/or the number of elements of a final quantity. In the model, this is expressed by the range of potential cardinalities. Common range data in the feature catalogue are: 1:1 Occurs precisely once. 1:1 Occurs once or more often. 0:1 Occurs never or once.	**cardinality**

	0:1 Occurs never or often.	
Karten-geometrieobjekt	Map geometry objects are thematic objects, which when derived for a certain map scale have changed their geometric form and/or position for reasons of cartographic generalisation.	**map geometry object**
Klasse	A class is a term used in object-oriented modelling and describes a number of objects that share the same attributes, methods, relations and the (dynamic) behaviour. descriptor of a set of objects that share the same attributes, operations, methods, relationships, and behaviour NOTE A class represents a concept within the system being modelled. Depending on the kind of model, the concept may be real-world (for an analysis model), or it may also contain algorithmic and computer implementation concepts (for a design model). A classifier is a generalization of class that includes other class-like elements, such as data type, actor and component. NOTE A class may use a set of interfaces to specify collections of operations it provides to its environment.	**class**
Kodierung	Encoding is the representation of information (data, objects) in a (electronically readable) encoding system; the inverse representation is decoding	**encoding**
konzeptuelles Modell	A conceptual model represents the real world in terms of concrete technical themes. model that defines the concepts of a universe of discourse	**conceptual model**
konzeptuelles Schema	The conceptual schema describes the conceptual model using a formal language. schema of a conceptual model A conceptual schema classifies objects into types and classes, identifying types of objects according to their properties and associations between types of objects.	**conceptual schema**
Metadaten	Metadata are data on data. They describe geodata in terms of user-relevant aspects for evaluating the suitability of the data and access to the same. ISO differentiates between some 400 optional, obligatory and absolutely obligatory metadata elements. data describing and documenting data	**metadata**
Metadaten-katalog	A metadata catalogue is a catalogue containing descriptive data (metadata). For each database, it contains specific details regarding the content, representation, expansion (both geometric and temporal), the spatial reference, quality and responsible institution on the basis of which a user can assess availability and suitability of the geodata records for his own purposes.	**metadata catalogue**

Metaobjekt-klasse	Metaclasses are defined as a basis from which thematic objects are instanced. A spatially-referenced metaclass (GF_FeatureType from ISO 19109) is used for modelling the basic classes.	**metaclass**
Methode	A method is a function relating to an object. It affects only this object and/or its properties (attributes, geometries and relations).	**method**
Modell	A model is a simplified pictorial or mathematical representation of structures and of the behaviour of complex situations in the real world. It is used to solve certain tasks that are impossible or impractical to solve in their original format. model abstraction of some aspects of reality	**model**
Modellierungs-sprache	Conceptual schema language provides illustrative and/or textual elements for describing a model. For modelling in the AFIS-ALKIS-ATKIS technical area, the Unified Modeling Language (UML) is used in accordance with ISO19103. formal language based on a conceptual formalism for the purpose of representing conceptual schemas EXAMPLE UML, EXPRESS, IDEFIX NOTE A conceptual schema language may be lexical or graphical.	**conceptual schema language**
Normen	De-jure standards standardise diverse areas of human activity. One type of de-jure standards is ISO: Documents created by members of the International Organization for Standardization (ISO) in so-called Technical Committees (TC) as part of a multi-stage development process. The TC211 “Geographic information/Geomatics” is responsible for geoinformation (see http://www.isotc211.org/). These documents pass through various stages of maturity. The end stage is that of the “International Standard”. For more information: See http://www.iso.ch/.	**de-jure standards**
Nutzer-spezifische Bestandsdaten-aktualisierung	Opertion used to udate secondary databases with difference-only data	**user-specific upting of secondary databases**
Objekt	An object (instance or class) is a tangible or intangible object of technical reality, which is uniquely identifiable and limited through abstraction of its relevant properties. This includes its condition and its behaviour. → Thematic object a discrete entity with a well-defined boundary and identity	**object**

	that encapsulates state and behaviour; an instance of a class	
Objektart	Objects are classified according to various features types. For each feature type, all permitted properties are stipulated in the feature catalogue (type level). These stipulations also apply unconditionally for all characteristics (instance level), i.e. the individual objects of this type. Each Object belongs to precisely one feature type. class of real world phenomena with common properties EXAMPLE The phenomenon 'Eiffel Tower' may be classified with other similar phenomena into a feature type 'tower'. NOTE In a feature catalogue, the basic level of classification is the feature type.	**feature type**
Objektarten-katalog	The feature catalogue lists the data elements with their stipulations modelled on the basis of the AFIS-ALKIS-ATKIS application schema for all feature types. catalogue containing definitions and descriptions of the feature types, feature attributes, and feature relationships occurring in one or more sets of geographic data, together with any feature operations that may be applied	**feature catalogue**
Objektbehälter	The feature version container for feature versions forms a technical bracket around the various versions of an object, which pass through it during the course of its life. By "bracketing" the versions within a feature version container for feature versions, the technical appreciation of the object remains in place.	**container for feature versions -**
Objekt-identifikator	→ Identifier	**object identifier**
Objekt-orientierung	The basis of object orientation, which is used for both the object-oriented modelling of systems and processes, object-oriented programming and also for object-oriented database management systems, is the abstraction of reality into objects, classes and relations. Object-orientation is therefore a method (concept, language) used to model facts for which all the required information (data and methods) is recorded as encapsulated objects that can communicate with each other.	**object orientation**
Objekt-strukturierung	Object structuring states that the facts modelled in an application schema exist in the structure of objects and are assigned by objects. Unlike object orientation, object structured modelling does not describe the behaviour of an object that is represented by its methods.	**object structuring**
Präsentations-objekt	Presentation objects are spatially-related elementary objects that supplement the features by details on representing text and signatures. All texts and	**presentation object**

	signatures that cannot be not fully automatically generated and positioned for a particular target scale are defined. Presentation objects should be defined in the feature catalogue on which the area based (e.g. ATKIS-basis-OK).	
Primärnachweis	The primary database is the original database managed by the appropriate relevant location (data manager).	**primary database**
Protokollobjekt	A protocol object is used to transfer protocol information.	**protocol object**
Prozess	A process transfers a source database to a target database. The following terms are used to describe the processes (operations and methods) textual, form-type description and pseudocodes . The “Processes in ALKIS” contain the definitions and descriptions of the methods and operations and also the process feature types used to control the processes.	**process**
Pseudocode	The pseudocode is a term used to describe a process. It contains the description for the processing stages of an operation annotated as follows: "objektart.methode (parameter)".	**pseudocode**
PunktLinienThema	A point and line these as defined by modelling comprises the option of grouping features in such a way as to be able to make common use of the geometries. As a result, the oppositely positioned lines and points are reciprocally separated and unite to form non-redundant geometries. Crossing lines do not result in reciprocal separation. Overlapping areas do not separate out into the smallest-possible part areas.	**point and line theme**
Qualifizierungs-prozess	The qualifying process transfers the collection data (output data) to the updating data (target data). This is a method of quality assurance and ensures that the updating data satisfy the quality requirements.	**qualifying process**
Raumbezug	The spatial reference is the geometric (position and form of the object) and/or the topological (positional relationship between objects) description of an object and thus creates the reference of the object to a spatial cutout in the earth.	**spatial reference**
Raumbezugs-grundform	Geometrical and topological primitives are provided by ISO 19107 *Spatial schema* for use in application schemas, pre-defined "Geometric Objects" (GM_Object) and "Topological objects" (TP_Object), that are described as UML classes. Geometrical and topological primitives are usually carried as attribute values of the objects.	**geometrical and topological primitives**
Relation	The term *“Relation”* means a general semantic connection between model elements. *Relation* is the generic term covering the terms *Association, Generalisation/Specialisation, Dependence* and	**relation**

	Realisation/Refining.	
Schema	A schema is a pictorial representation of the essential content of a fact. It is the result of the pictorial and/or textual description of a model using a (standardised) conceptual schema language. formal description of a model	**schema**
Sekundär-nachweis	The secondary database contains a copy of the entire primary database of parts of same, which are continuously being updated. The secondary database is revised through the user-related update of the primary database (NBA).	**secondary database**
Signaturen-katalog	A portrayal catalogue contains regulations according to which the outputs of geodata defined in the output catalogue are portrayed in dependence of their feature type, certain attributes/attribute values, certain reference conditions and/or the values to be calculated and the description of all existing portrayals. It is modified to the respective target scale.	**portrayal catalogue**
Standard	A de-facto standard is a widely accepted and applied instrument. It is usually generated by only *one* institution, i.e. it has no international body. The binding nature of a de-facto standard often remains within an individual organisation. A de-facto standard is not officially published as an international document, as is the case with a de-jure standard. There is no regular creation process (as is the case with de-jure standards, e.g. DIN, ISO or CEN). → AdV Standard	**de-facto standard**
Standard-ausgaben	Standard outputs cover normal cases of use (also within the sense of standard products of the AdV). Output products of AFIS-ALKIS-ATKIS data, which satisfy the normal or “standardised“ requirements for the relevant databases. These are stipulated through the definition of unified selection and filter criteria. Examples of standard outputs for ALKIS are the real estate map, the land parcel and ownership database and the real estate map containing land parcel and ownership details.	**standard output**
Subschema	→ Basic schema	**subschema**
Transferprozess	*See GeoInfoDok, 3.3.9.6*	**transfer process**
URI	Uniform Resource Identifier Character chain that refers uniquely to one resource (name, file etc.) The location of the resources is not restricted (www, LAN, …). URLs (Uniform Resource Locator) and URNs (Uniform Resource Name) are part quantities of URIs.	**URI (Uniform Resource Identifier) generic set of all names/addresses that are short strings that refer to**

		resources
Versionierung	Versioning is the temporally arranged change to features through updating. The core point of the version concept is the consideration that besides other information, each feature also carries a lifetime interval (consisting of creation and expiry date). → Versioning schema	**versioning**
Versionierungs-schema	The versioning schema is part of the conceptual basic schema and describes aspects of the temporal change to the features through updatings (see Fig. 3.4-1). → Feature version container for feature versions → Versioning	**versioning schema**
Vorgang	See 3.7.2.1	**operation**
XML-Schema	The XML schema is the lexical description of an application schema based on XML (Extensible Markup Language). On the basis of the structures stipulated in the XML schema, XML documents can be created for the exchange of data. See Item 10.1.1 and http://www.w3.org/TR/xmlschema-0/.	**XML schema**
Zeitstempel	The time stamp consists of creation date and time, which are taken from the "lifetime interval" attribute. It is intended as an addition to the object identifier and should enable specific identification of object versions on updating. See also 10.1.3.2.	**time stamp**
zusammen-gesetztes Objekt (ZUSO)	Composed objects are formed in order to create the correlation between any number and combination of semantically associated, spatially-referenced elementary objects, non-spatially referenced elementary objects or composed objects. However, a composed object must have at least one object as a component. → See also "elementary objects"	**composed object or complex object**

12.2 List of Abbreviations Used

Abbreviation	Long version
AdV	Working Committee of the Surveying Authorities of the States of the Federal Republic of Germany
AFIS	Official fixed point information system
ALB	Automated real estate register
ALK	Automated real estate map
ALKIS	Official real estate cadastre information system
ATKIS	Official topographic cartographic information system
ATS	Abstract Test Suite
BKG	Federal Agency for Cartograpy and Geodesy
CD	Commitee Draft
CSL	Conceptual Schema Language
DB	Database
DBM	Digital image model

DGM	Digital terrain model
DLM	Digital landscape model
DOP	Digital orthophoto
DTD	Document Type Definition
DTK	Digital topographic map
DXF	Data Exchange Format
FIS	Technical information system
GeoInfoDok	Documentation on the Modelling of Geoinformation of Official Surveying and Mapping
GIS	Geo information system
GML	Geography Markup Language
ID	Identifier
ISO	International Organization for Standardization
NAS	De-jure based data exchange interface
NBA	User-related update of primary databasee
NREO	Non-spatially referenced elementary object
OGC	Open GIS Consortium
OK	Feature catalogue
REO	Spatially-related elementary object
SK	Portrayal catalogue
TC	Technical Commitee
TK	Topographic map
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
UUID	Universally Unique Identifier
XML	Extensible Markup Language
ZUSO	Composed object

12.3 List of Figures

12.4 Status and version numbers of the documents

For the status and the version numbers for all the documents within the AAA-project see: http://www.adv-online.de/veroeffentlichungen/afis-alkis-atkis/geoinfodok-index-3-0.htm