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

AFIS
ALKIS
ATKIS

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

(GeoInfoDoc)

Main Document

Version 5.1

July 31, 2006

Edited by
the AdV-working group
Information and Communication Technology

Table of contents

1 Structure, content and objective

The GeoInfoDoc contains of a main document (this document), the AAA application schema and the corresponding derived documents (e.g. feature catalogues, NAS, coordinates referernc systems) The structure of this document gors in line with the numpering of chapter of teh derived catalogues. The following text just hints are given to the separate documents. The whole GeoInfoDoc in its actual and historical versions is published and maintained at www.adv-online.de. Some of the clauses, that are not published as separate documents, are treated just in this main document. The main documents containes besides a lot of explanations and modelling principles also important remarks about the specification of the NAS (see Section 10). The statements of this main document therefor are normative and this way they extend the specification of the AAA-application schema.

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.

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, versioning and migration

For economic reason, nation wide users and GIS vendors demand an accepted nation wide harmonised core data regarding the content and structure of the geo-base-data. The core data of AFIS, ALKIS and ATKIS shall be combined to just one core data representing all spatial data of the official surveying in Germany. **Grunddatenbestand 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.

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).

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.

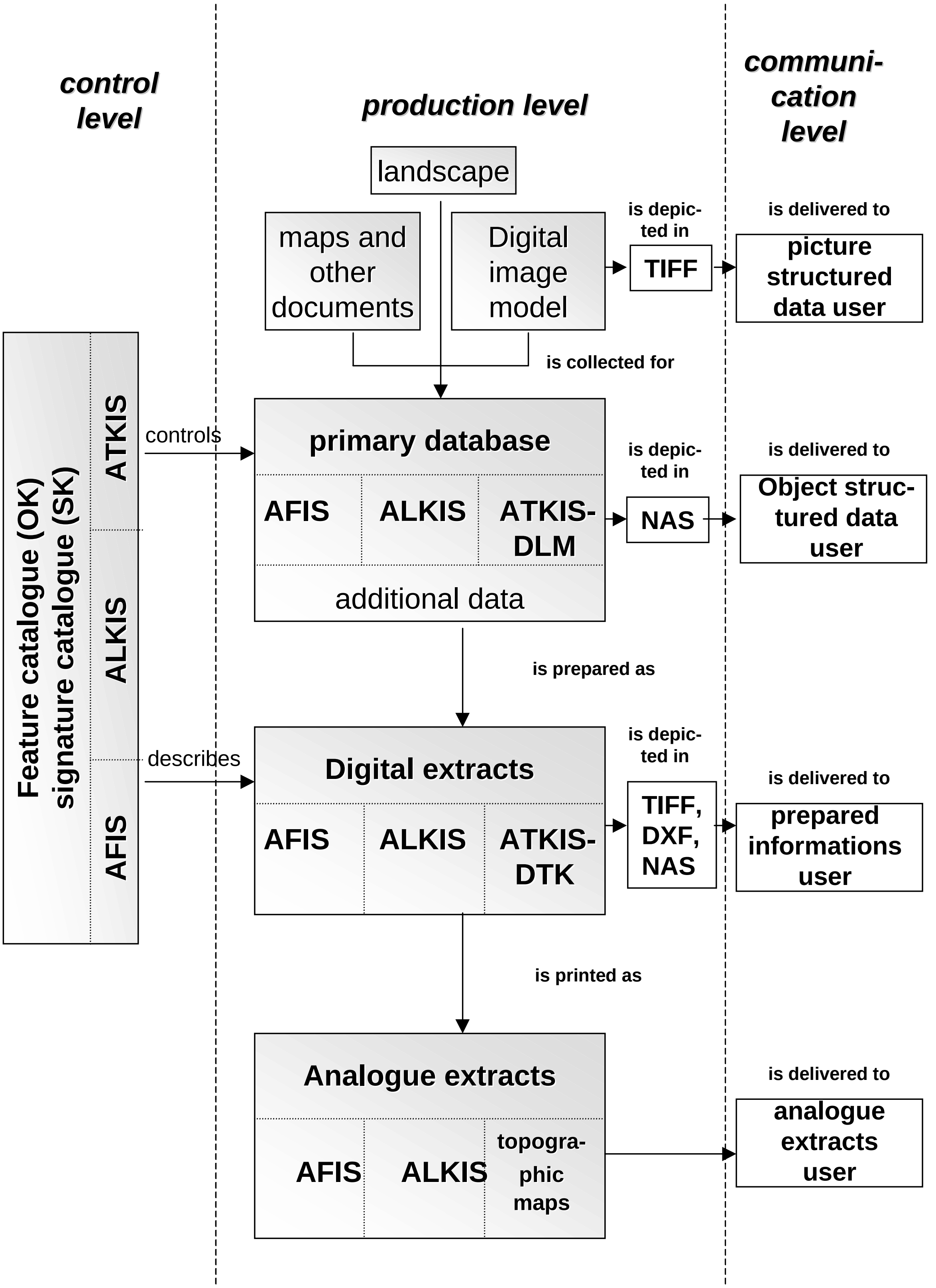


Figure 1: Common AFIS-ALKIS-ATKIS Reference Model

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 Geospatial 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 Open Geospatial 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.

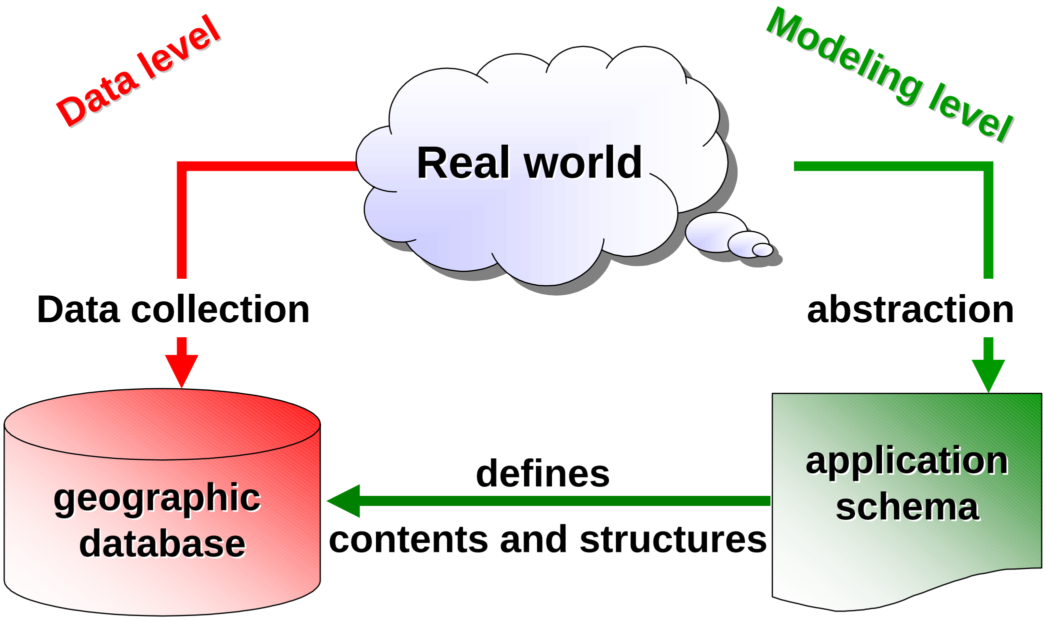


Figure 2: The role of the application schema

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 Open Geospatial-Consortium are used.

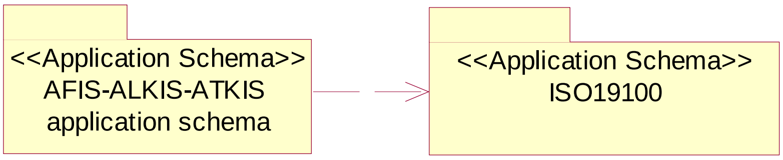


Figure 3: Dependency of the AFIS-ALKIS-ATKIS application schema on the structures standardised from ISO 19100

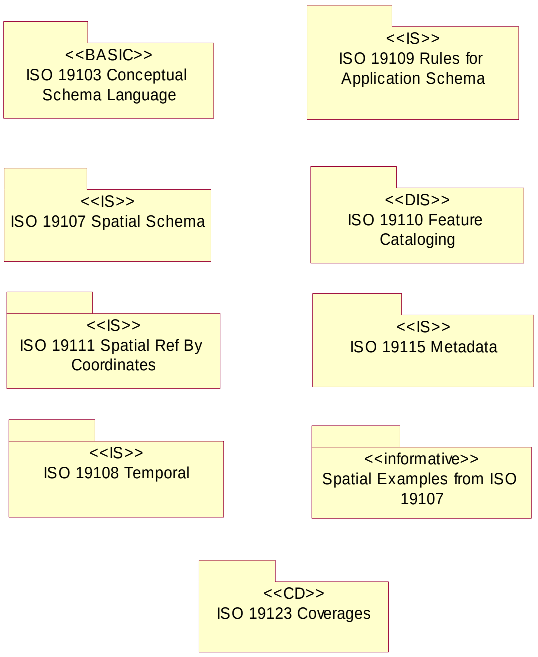

<<BASIC>>
ISO 19103 Conceptual Schema Language
<<IS>>
ISO 19109 Rules for Application Schema
<<IS>>
ISO 19107 Spatial Schema
<<DIS>>
ISO 19110 Feature Cataloging
<<IS>>
ISO 19111 Spatial Ref By Coordinates
<<IS>>
ISO 19115 Metadata
<<IS>>
ISO 19108 Temporal
<<informative>>
Spatial Examples from ISO 19107
<<CD>>
ISO 19123 Coverages

Figure 4: 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 NAS operations (Section 10.2) and the AFIS-ALKIS-ATKIS output catalogue. Besides that there are some mor packages that are necessary for the specification of the data exchange (see Section 10).

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.

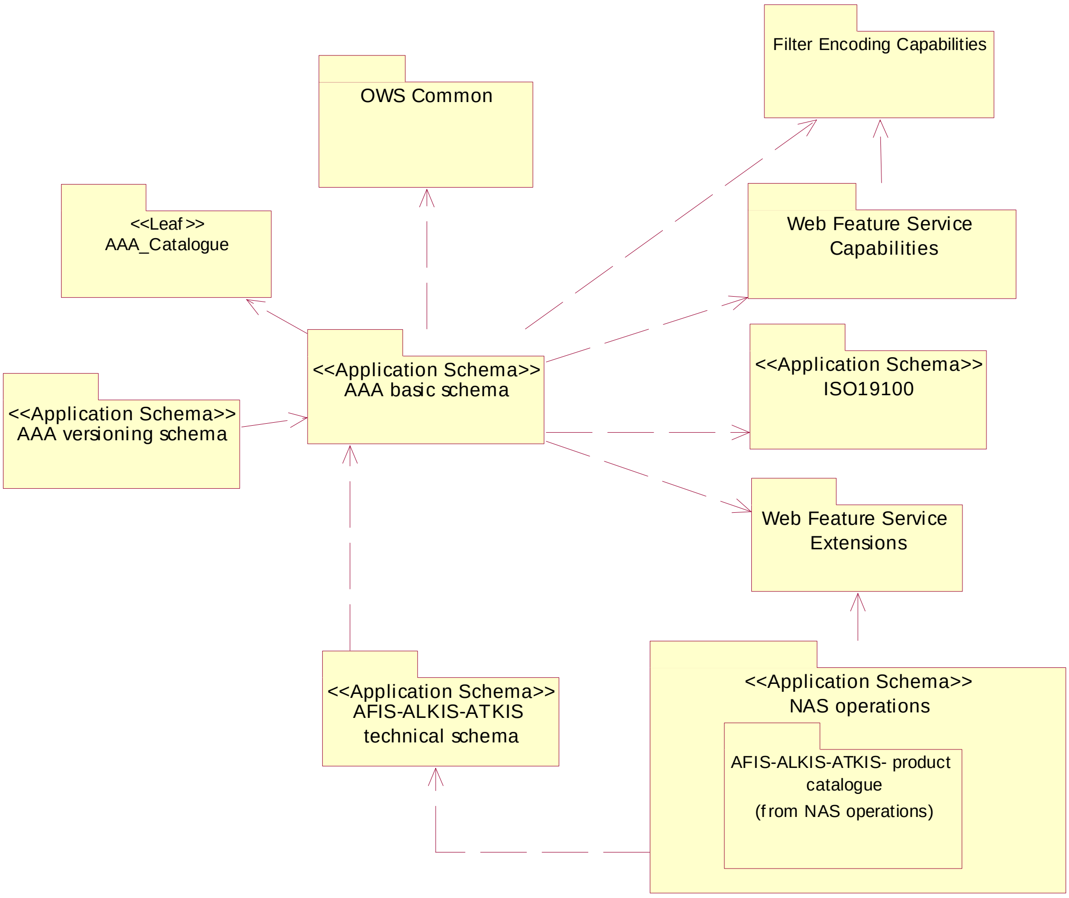


Figure 5: The components of the AFIS-ALKIS-ATKIS application schema

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.

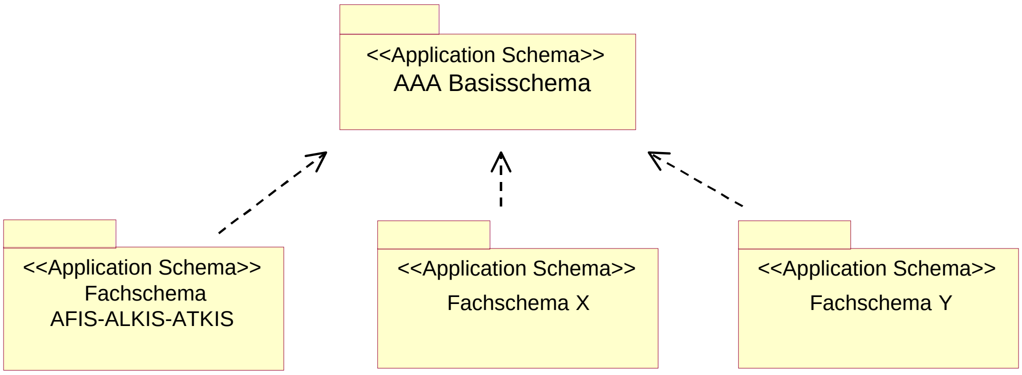


Figure 6: The basic schema as a basis for the modelling of application-specific thematic schemas (e.g. AFIS, ALKIS and ATKIS)

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

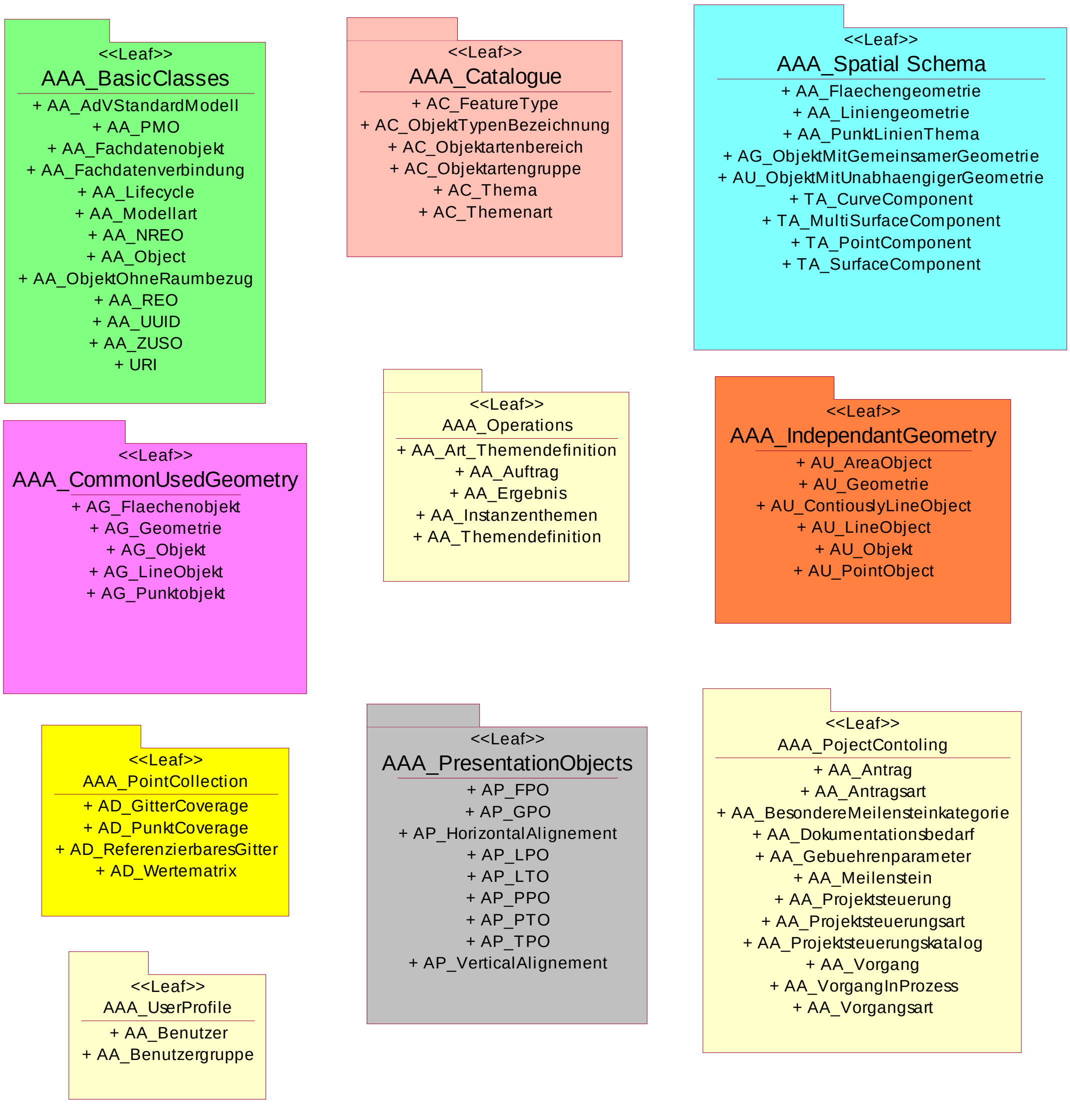


Figure 7: Components of the basic schema

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
- AP_Presentation

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.

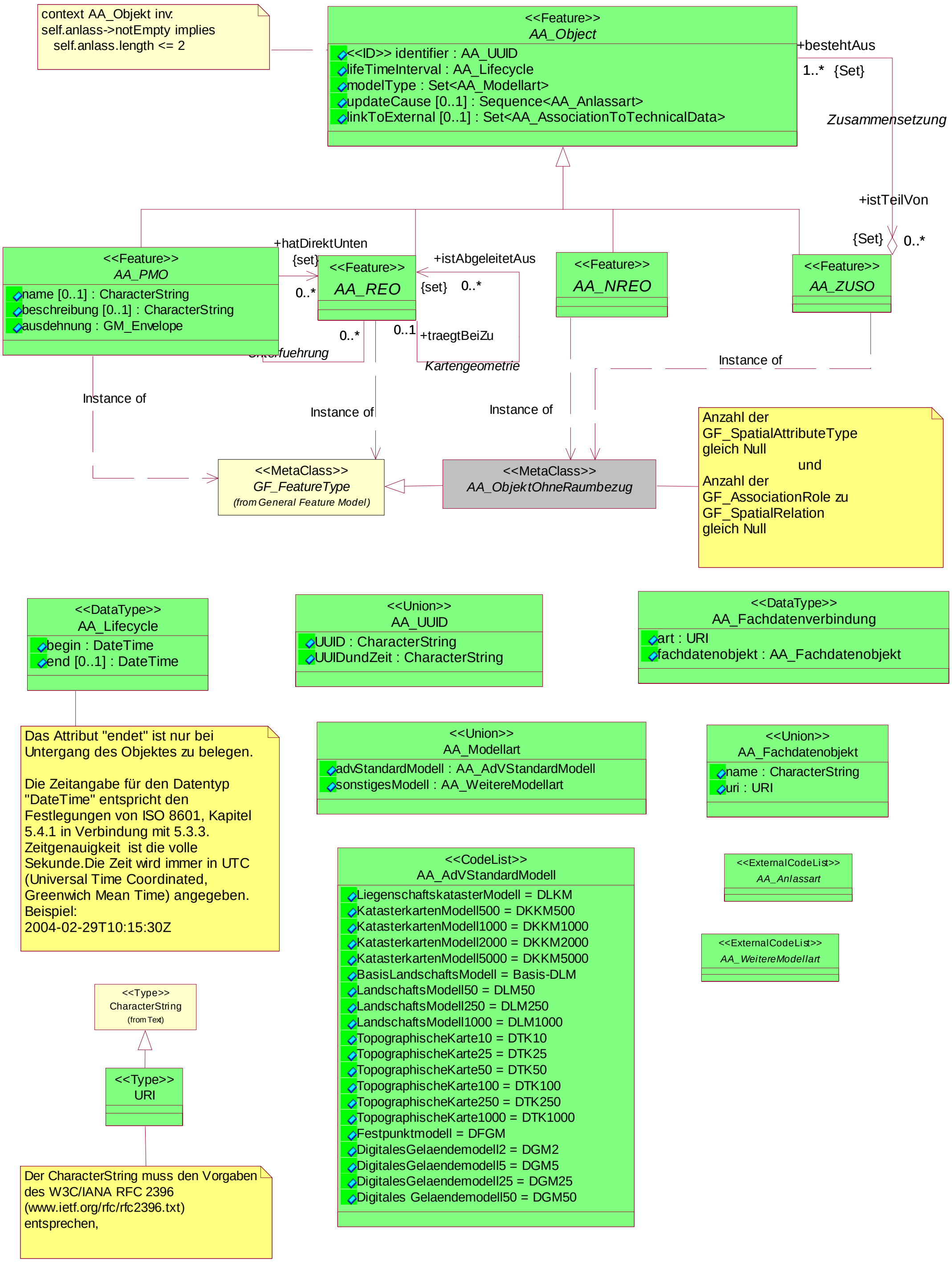

context AA_Objekt inv:
self.anlass->notEmpty implies
self.anlass.length <= 2
<<Feature>>
AA_Object
<<ID>> identifier : AA_UUID
lifeTimeInterval : AA_Lifecycle
modelType : Set<AA_Modellart>
updateCause [0..1] : Sequence<AA_Anlassart>
linkToExternal [0..1] : Set<AA_AssociationToTechnicalData>
+bestehtAus
1..* {Set}
Zusammensetzung
+istTeilVon
{Set} 0..*
<<Feature>>
AA_PMO
name [0..1] : CharacterString
beschreibung [0..1] : CharacterString
ausdehnung : GM_Envelope
+hatDirektUnten
{set}
0..*
<<Feature>>
AA_REO
+istAbgeleitetAus
{set} 0..*
0..* 0..1
+traegtBeiZu
Unterfuehrung
Kartengeometrie
<<Feature>>
AA_NREO
<<Feature>>
AA_ZUSO
Instance of
Instance of
Instance of
Instance of
<<MetaClass>>
GF_FeatureType
(from General Feature Model)
<<MetaClass>>
AA_ObjektOhneRaumbezug
Anzahl der
GF_SpatialAttributeType
gleich Null
und
Anzahl der
GF_AssociationRole zu
GF_SpatialRelation
gleich Null
<<DataType>>
AA_Lifecycle
begin : DateTime
end [0..1] : DateTime
<<Union>>
AA_UUID
UUID : CharacterString
UUIDundZeit : CharacterString
<<DataType>>
AA_Fachdatenverbindung
art : URI
fachdatenobjekt : AA_Fachdatenobjekt
Das Attribut "endet" ist nur bei
Untergang des Objektes zu belegen.
Die Zeitangabe für den Datentyp
"DateTime" entspricht den
Festlegungen von ISO 8601, Kapitel
5.4.1 in Verbindung mit 5.3.3.
Zeitgenauigkeit ist die volle
Sekunde.Die Zeit wird immer in UTC
(Universal Time Coordinated,
Greenwich Mean Time) angegeben.
Beispiel:
2004-02-29T10:15:30Z
<<Union>>
AA_Modellart
advStandardModell : AA_AdVStandardModell
sonstigesModell : AA_WeitereModellart
<<Union>>
AA_Fachdatenobjekt
name : CharacterString
uri : URI
<<CodeList>>
AA_AdVStandardModell
LiegenschaftskatasterModell = DLKM
KatasterkartenModell500 = DKKM500
KatasterkartenModell1000 = DKKM1000
KatasterkartenModell2000 = DKKM2000
KatasterkartenModell5000 = DKKM5000
BasisLandschaftsModell = Basis-DLM
LandschaftsModell50 = DLM50
LandschaftsModell250 = DLM250
LandschaftsModell1000 = DLM1000
TopographischeKarte10 = DTK10
TopographischeKarte25 = DTK25
TopographischeKarte50 = DTK50
TopographischeKarte100 = DTK100
TopographischeKarte250 = DTK250
TopographischeKarte1000 = DTK1000
Festpunktmodell = DFGM
DigitalesGelaendemodell2 = DGM2
DigitalesGelaendemodell5 = DGM5
DigitalesGelaendemodell25 = DGM25
Digitales Gelaendemodell50 = DGM50
<<ExternalCodeList>>
AA_Anlassart
<<ExternalCodeList>>
AA_WeitereModellart
<<Type>>
CharacterString
(from Text)
<<Type>>
URI
Der CharacterString muss den Vorgaben
des W3C/IANA RFC 2396
(www.ietf.org/rfc/rfc2396.txt)
entsprechen,

Figure 8: 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.

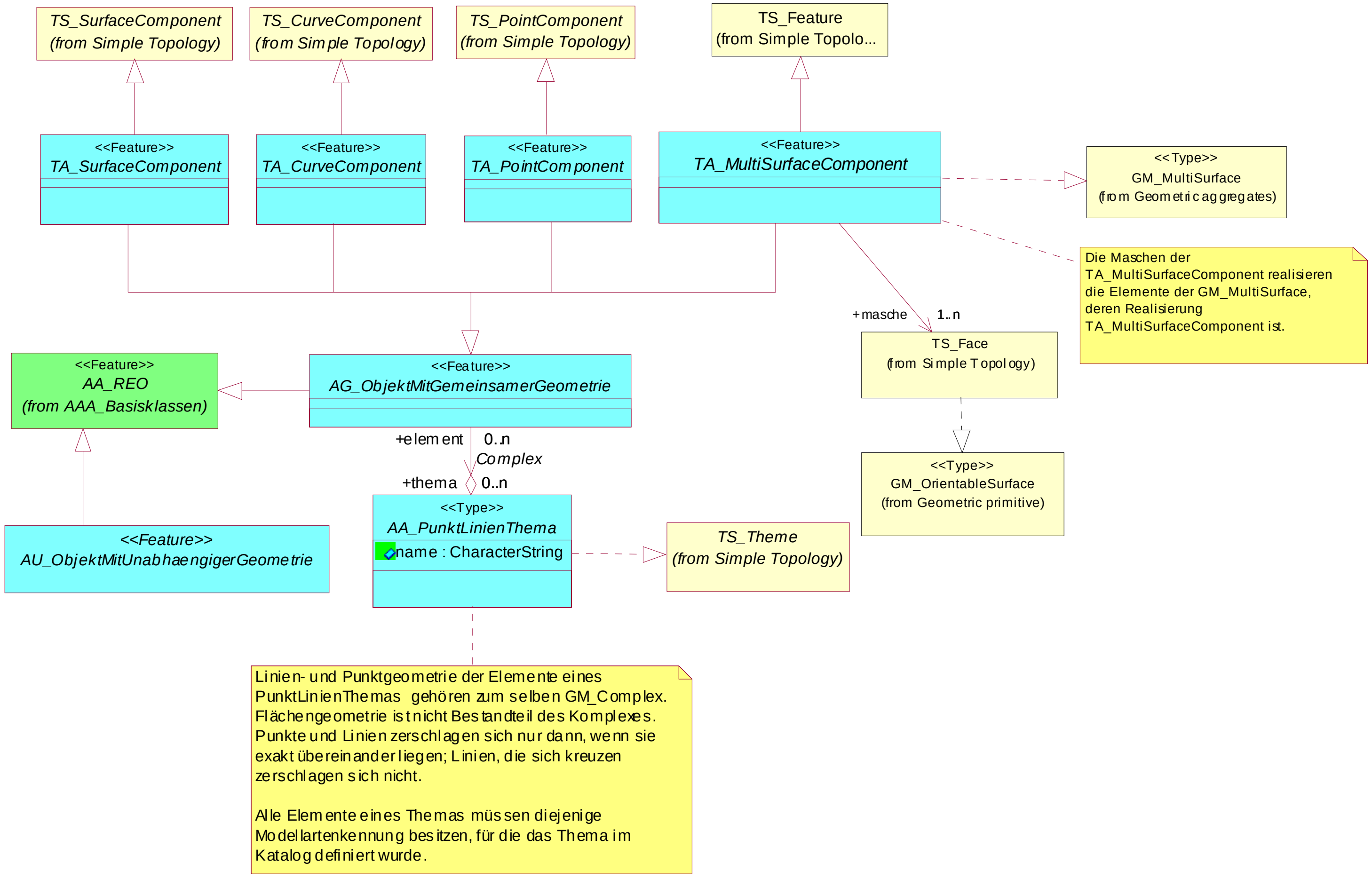
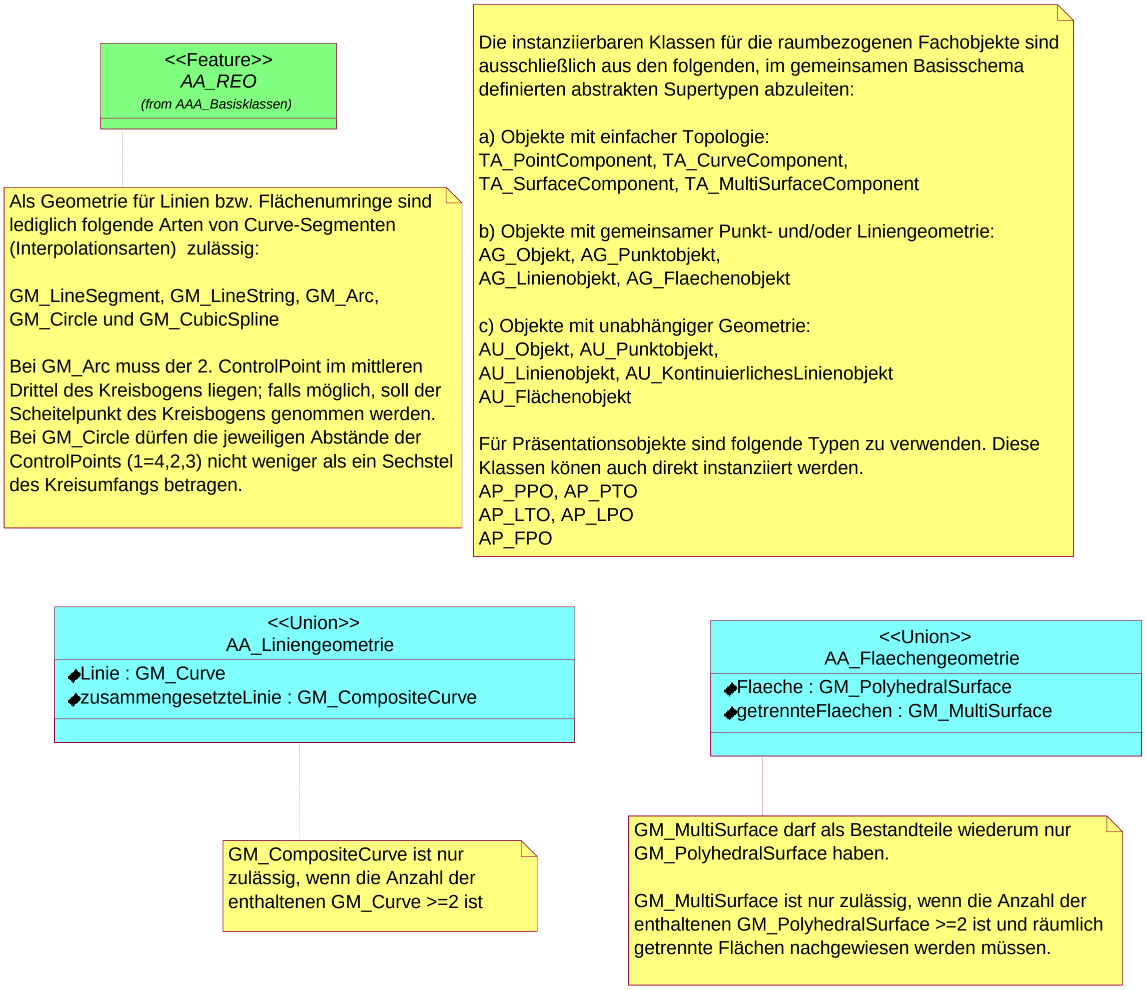

TS_SurfaceComponent
(from Simple Topology)
TS_CurveComponent
(from Simple Topology)
TS_PointComponent
(from Simple Topology)
TS_Feature
(from Simple Topolo...
<<Feature>>
TA_SurfaceComponent
<<Feature>>
TA_CurveComponent
<<Feature>>
TA_PointComponent
<<Feature>>
TA_MultiSurfaceComponent
<<Type>>
GM_MultiSurface
(from Geometric aggregates)
Die Maschen der TA_MultiSurfaceComponent realisieren die Elemente der GM_MultiSurface, deren Realisierung TA_MultiSurfaceComponent ist.
+masche
1..n
TS_Face
(from Simple Topology)
<<Type>>
GM_OrientableSurface
(from Geometric primitive)
<<Feature>>
AA_REO
(from AAA_Basisklassen)
<<Feature>>
AG_ObjektMitGemeinsamerGeometrie
+element
0..n
Complex
+thema
0..n
<<Type>>
AA_PunktLinienThema
name : CharacterString
TS_Theme
(from Simple Topology)
<<Feature>>
AU_ObjektMitUnabhaengigerGeometrie
Linien- und Punktgeometrie der Elemente eines PunktLinienThemas gehören zum selben GM_Complex. Flächengeometrie ist nicht Bestandteil des Komplexes. Punkte und Linien zerschlagen sich nur dann, wenn sie exakt übereinander liegen; Linien, die sich kreuzen zerschlagen sich nicht.
Alle Elemente eines Themas müssen diejenige Modellartenkennung besitzen, für die das Thema im Katalog definiert wurde.

Figure 9: Summarizing representation of the required supplements for AFIS-ALKIS-ATKIS to the standardised Spatial Schema



Die instanziierbaren Klassen für die raumbezogenen Fachobjekte sind ausschließlich aus den folgenden, im gemeinsamen Basisschema definierten abstrakten Supertypen abzuleiten:

a) Objekte mit einfacher Topologie:
TA_PointComponent, TA_CurveComponent,
TA_SurfaceComponent, TA_MultiSurfaceComponent

b) Objekte mit gemeinsamer Punkt- und/oder Liniengeometrie:
AG_Objekt, AG_Punktobjekt,
AG_Linienobjekt, AG_Flaechenobjekt

c) Objekte mit unabhängiger Geometrie:
AU_Objekt, AU_Punktobjekt,
AU_Linienobjekt, AU_KontinuierlichesLinienobjekt
AU_Flächenobjekt

Für Präsentationsobjekte sind folgende Typen zu verwenden. Diese Klassen könen auch direkt instanziert werden.
AP_PPO, AP_PTO
AP_LTO, AP_LPO
AP_FPO

Als Geometrie für Linien bzw. Flächenumringe sind lediglich folgende Arten von Curve-Segmenten (Interpolationsarten) zulässig:

GM_LineSegment, GM_LineString, GM_Arc, GM_Circle und GM_CubicSpline

Bei GM_Arc muss der 2. ControlPoint im mittleren Drittel des Kreisbogens liegen; falls möglich, soll der Scheitelpunkt des Kreisbogens genommen werden. Bei GM_Circle dürfen die jeweiligen Abstände der ControlPoints (1=4,2,3) nicht weniger als ein Sechstel des Kreisumfangs betragen.

GM_CompositeCurve ist nur zulässig, wenn die Anzahl der enthaltenen GM_Curve >=2 ist

GM_MultiSurface darf als Bestandteile wiederum nur GM_PolyhedralSurface haben.

GM_MultiSurface ist nur zulässig, wenn die Anzahl der enthaltenen GM_PolyhedralSurface >=2 ist und räumlich getrennte Flächen nachgewiesen werden müssen.

Figure 10: Restrictions in respect of geometry and instantiable classes

The allowed interpolation types for lines and surface boundaries are limited to GM_LineSegment, GM_LineString, GM_Arc, GM_Circle und GM_CubicSpline. These elements are principally valid for all thematic schemas based on the basic schema, if it is no other restriction formulated there. The ALKIS thematic scheme do not allow any GM_CubicSplines.

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.

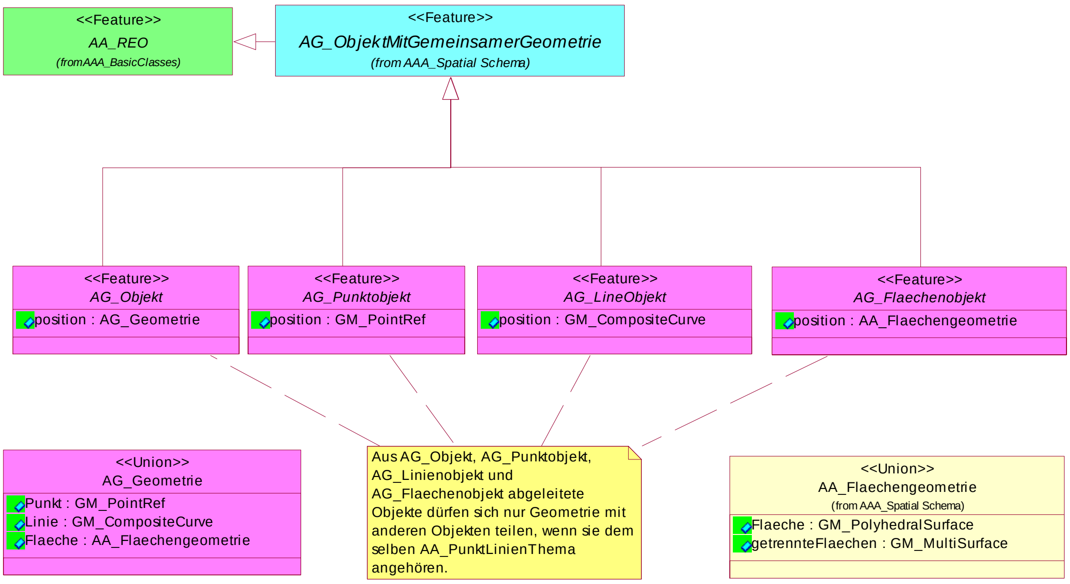

<<Feature>>
AA_REO
(fromAAA_BasicClasses)
<<Feature>>
AG_ObjektMitGemeinsamerGeometrie
(from AAA_Spatial Schema)
<<Feature>>
AG_Objekt
position : AG_Geometrie
<<Feature>>
AG_Punktobjekt
position : GM_PointRef
<<Feature>>
AG_LineObjekt
position : GM_CompositeCurve
<<Feature>>
AG_Flaechenobjekt
position : AA_Flaechengeometrie
<<Union>>
AG_Geometrie
Punkt : GM_PointRef
Linie : GM_CompositeCurve
Flaeche : AA_Flaechengeometrie
Aus AG_Objekt, AG_Punktobjekt, AG_Linienobjekt und AG_Flaechenobjekt abgeleitete Objekte dürfen sich nur Geometrie mit anderen Objekten teilen, wenn sie dem selben AA_PunktLinienThema angehören.
<<Union>>
AA_Flaechengeometrie
(from AAA_Spatial Schema)
Flaeche : GM_PolyhedralSurface
getrennteFlaechen : GM_MultiSurface

Figure 11: 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).

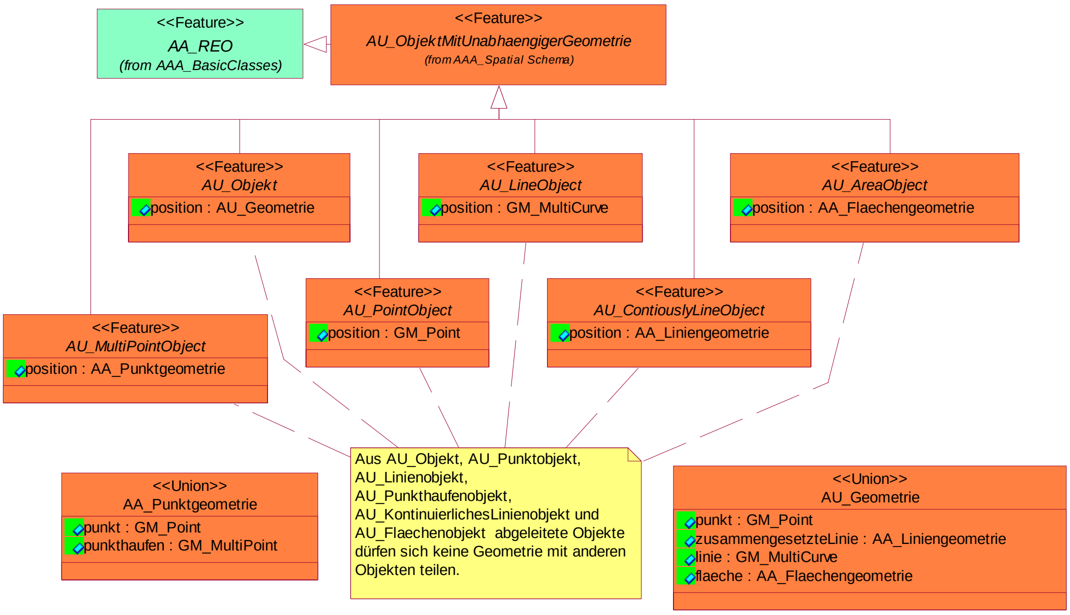
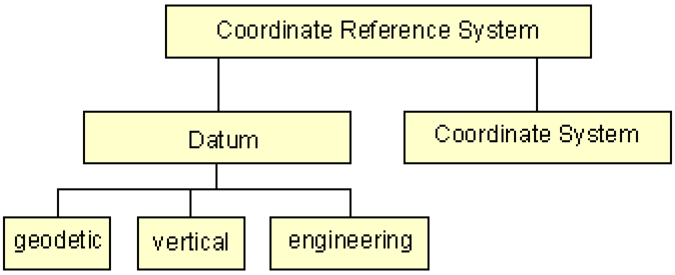

<<Feature>>
AA_REO
(from AAA_BasicClasses)
<<Feature>>
AU_ObjektMitUnabhaengigerGeometrie
(from AAA_Spatial Schema)
<<Feature>>
AU_Objekt
position : AU_Geometrie
<<Feature>>
AU_LineObject
position : GM_MultiCurve
<<Feature>>
AU_AreaObject
position : AA_Flaechengeometrie
<<Feature>>
AU_PointObject
position : GM_Point
<<Feature>>
AU_ContiouslyLineObject
position : AA_Liniengeometrie
<<Feature>>
AU_MultiPointObject
position : AA_Punktgeometrie
<<Union>>
AA_Punktgeometrie
punkt : GM_Point
punkthaufen : GM_MultiPoint
Aus AU_Objekt, AU_Punktobjekt, AU_Linienobjekt, AU_Punkthaufenobjekt, AU_KontinuierlichesLinienobjekt und AU_Flaechenobjekt abgeleitete Objekte dürfen sich keine Geometrie mit anderen Objekten teilen.
<<Union>>
AU_Geometrie
punkt : GM_Point
zusammengesetzteLinie : AA_Liniengeometrie
linie : GM_MultiCurve
flaeche : AA_Flaechengeometrie

Figure 12: 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.

According to ISO 19111 *(Spatial Referencing by Coordinates)* a coordinate reference system contains of two componets, the datumg and the coordinate system (see figure besides).



The datum is the physical part of a CRS that defines reference to the earth by the zero point, orientation of the coordinate axis and the scale. A datum can be a geodetic datum, a verical datum or an engeneering (local) datum. Examples for a geodetic datum is the German Hauptdreiecksnetz (DHDN), also called “Potsdam-Datum“, ort he European Terrestical Referenc System (ETRS 89).

The coordinate system ist he mathematical part of a CRS that is defined by rules how to apply coordinates to a geometry, for example a geodetic fix point. The coordinates of a geometry can be cartesic coordinates (X,Y,Z), ellopsoidic corrdinates (latitude, longitute, and ellipsoidic height) or projected coordinates (Gauß-Krueger-projection, UTM).

Besides the CRS for two dimensional and three dimensional spatial representation there are necessary separate coordinate reference systems for height or height coordinates (e.g. NN height). 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). According to ISO 19111 it is possible to introduce combined CRS. Nevertheless, for the feature class „AX_PointLocation“ combined coordinate referernce systems are **not allowed** in AFIS-ALKIS-ATKIS.

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

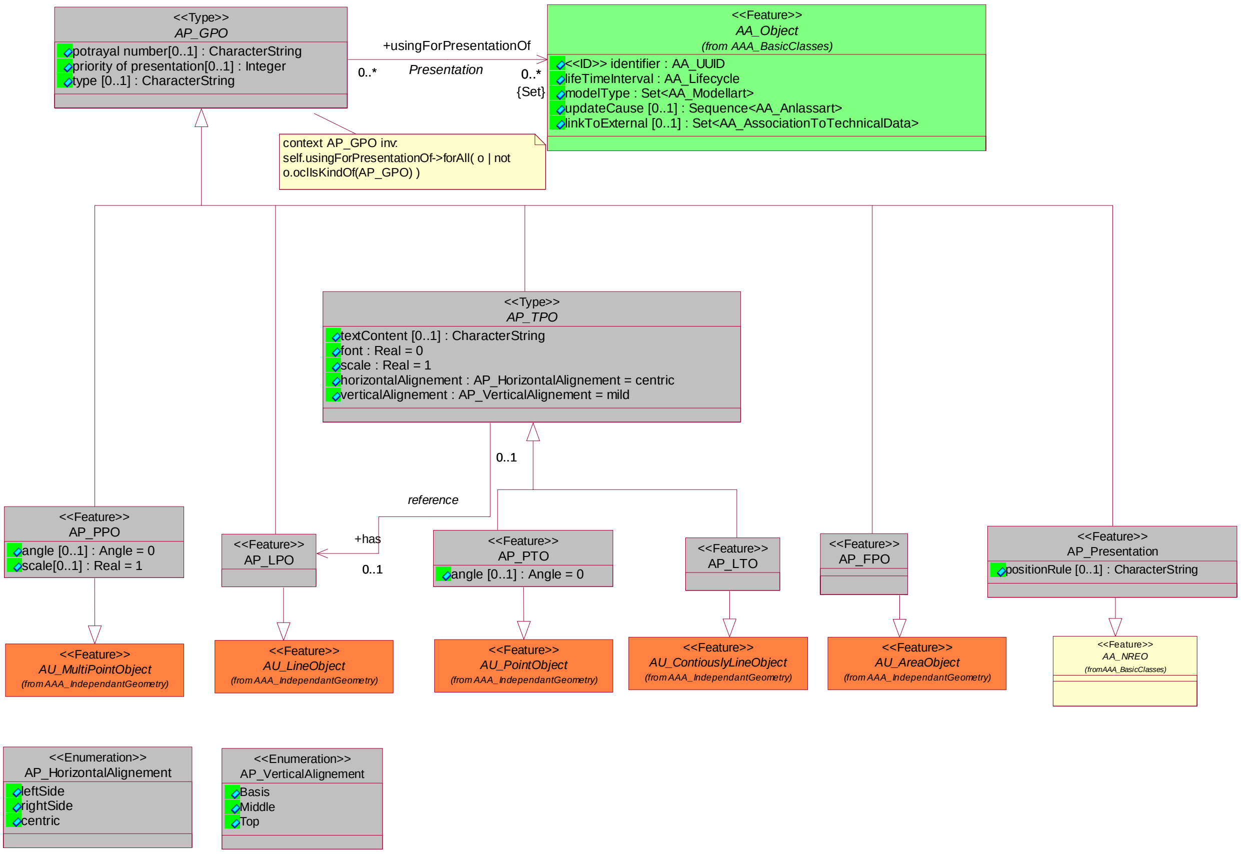
The complete description of features comprises the following components

- 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.



<<Type>>
AP_GPO
potrayal number[0..1] : CharacterString
priority of presentation[0..1] : Integer
type [0..1] : CharacterString
+usingForPresentationOf
Presentation
0..*
0..*
{Set}
<<Feature>>
AA_Object
(from AAA_BasicClasses)
<<ID>> identifier : AA_UUID
lifeTimeInterval : AA_Lifecycle
modelType : Set<AA_Modellart>
updateCause [0..1] : Sequence<AA_Anlassart>
linkToExternal [0..1] : Set<AA_AssociationToTechnicalData>
context AP_GPO inv:
self.usingForPresentationOf->forAll(o | not o.oclIsKindOf(AP_GPO))
<<Type>>
AP_TPO
textContent [0..1] : CharacterString
font : Real = 0
scale : Real = 1
horizontalAlignement : AP_HorizontalAlignement = centric
verticalAlignement : AP_VerticalAlignement = mild
0..1
reference
+has
0..1
<<Feature>>
AP_PPO
angle [0..1] : Angle = 0
scale[0..1] : Real = 1
<<Feature>>
AP_LPO
<<Feature>>
AP_PTO
angle [0..1] : Angle = 0
<<Feature>>
AP_LTO
<<Feature>>
AP_FPO
<<Feature>>
AP_Presentation
positionRule [0..1] : CharacterString
<<Feature>>
AU_MultiPointObject
(from AAA_IndependantGeometry)
<<Feature>>
AU_LineObject
(from AAA_IndependantGeometry)
<<Feature>>
AU_PointObject
(from AAA_IndependantGeometry)
<<Feature>>
AU_ContiouslyLineObject
(from AAA_IndependantGeometry)
<<Feature>>
AU_AreaObject
(from AAA_IndependantGeometry)
<<Feature>>
AA_NREO
(fromAAA_BasicClasses)
<<Enumeration>>
AP_HorizontalAlignement
leftSide
rightSide
centric
<<Enumeration>>
AP_VerticalAlignement
Basis
Middle
Top

Figure 13: 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*.

Presentation in the map

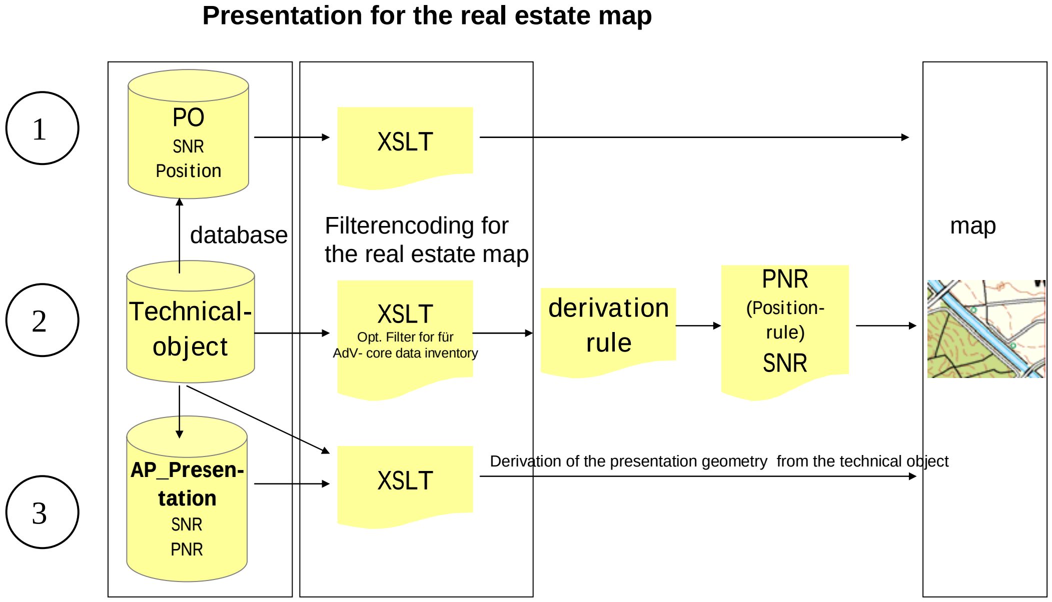

Presentation for the real estate map
1
2
3
PO
SNR
Position
database
Technical-object
AP_Presen-tation
SNR
PNR
XSLT
Filterencoding for the real estate map
XSLT
Opt. Filter for für AdV- core data inventory
XSLT
derivation rule
PNR
(Position-rule)
SNR
Derivation of the presentation geometry from the technical object
map

Figure 14: Presentation the real estate map

(1) <u>Presentation in the data in primary database</u>

Presentation objects will be built for all portrayals in form of text, line, symbol, area, which can not be automatically generated and presented for a special map scale. The specific portrayal number, which is presented by a concrete derivation rule, also the position number (PNR), which presents a concrete position rule can be optionally stored in the presentation object. Presentation objects are also built, if the presentation deviates from the standard presentation, stipulated in the portrayal catalogue.

(2) <u>Presentation with derivation rule and position rule</u>

Portrayals of a technical object in form of text, line, symbol, area will be placed at a define position (standard position) by using the filter encoding and a concrete derivation rule, which represents a concrete portrayal number and position rules, which activate a concrete number of position (see chapter 7.3, ALKIS-SK). In this case a presentation object is not created in the data in primary database. The technical information, which will be presented, is taken from the attribute type of the technical feature type. This way is the standard case, which is not always the most effective way.

(3) <u>Presentation with stored derivation rule and position rule</u>

In order to increase the performance of the standard presentation the concrete portrayal number and also the number of position, which is used to present a technical object, will be stored at a certain time (updating, first insert) under the referenced presentation object AP_Presentation (NREO). The advantage is, that the geometry is redundancy-free, because the presentation geometry of the feature type AP_Presentation is derived by the technical feature type. The presentation will be quickly executed by using the filter encoding in connection with the derivation of the presentation geometry from the technical feature type und the stored rules, portrayal number, number of position. In ALKIS the feature type AP_Presentation is exclusively used to handle the following changes in the real estate map:

- Suppression a presentation in the real estate map
- Creation of a specific presentation of symbols in the real estate map, e. g. arrangement of sysmbols in an area.

Presentation of a real estate description

The presentaiton of the information concerning the cadastral content, e.g. parcel certification/owner certification, is automatically derived on the fly by using Filter Encoding in order to create the corresponding output data. The output data are presented by using a specific derivation rule. The corresponding positions of the text are not specificly described in the portrayal catalogue, but are shown in a concrete sample (see chapter 7.3, ALKIS-SK)

The non spatial feature type AP_Presentaion is not applied in this context. Additionally there is no creation of presentaion objects in the primary data base.

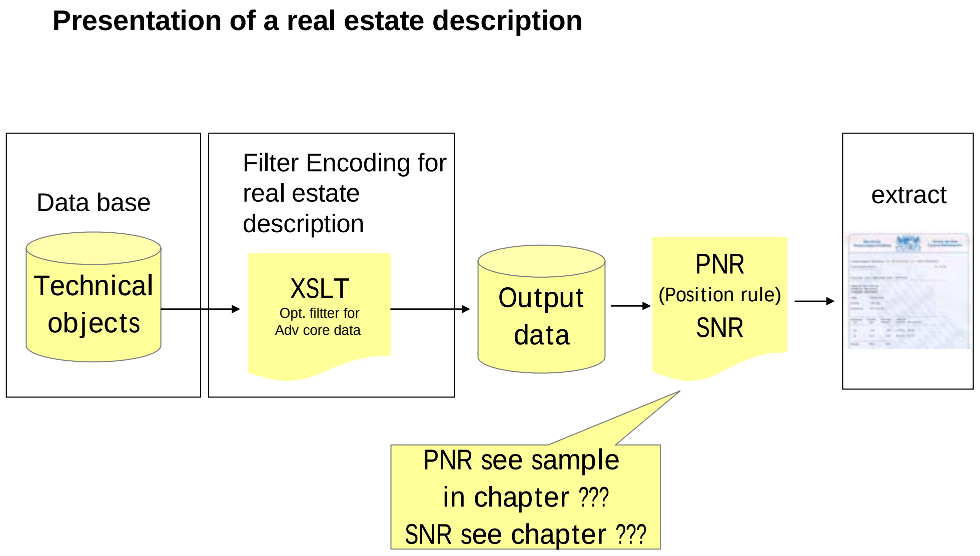

Presentation of a real estate description
Data base
Technical objects
Filter Encoding for real estate description
XSLT
Opt. filtter for Adv core data
Output data
PNR
(Position rule)
SNR
extract
PNR see sample in chapter ???
SNR see chapter ???

Figure 15: Presentation of a real estate description

Creation using presentation objects and AP_Presentation for the data in primary data base

In order to get a performant presentation of the technical objects in a cadastral extract, appropriate presentation rules must be defined in time of collecting or updating the data. This way the following three cases have to be destinguished.

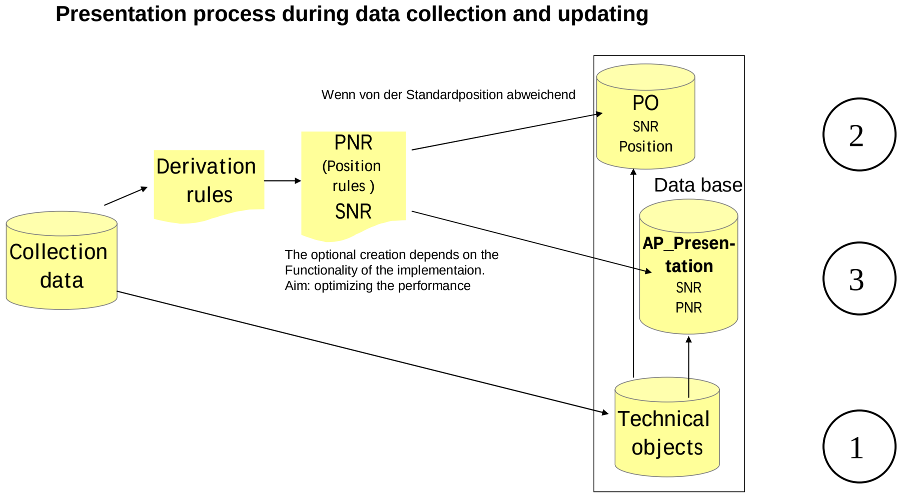

Presentation process during data collection and updating
Wenn von der Standardposition abweichend
PO
SNR
Position
Data base
Derivation rules
PNR
(Position rules)
SNR
Collection data
The optional creation depends on the
Functionality of the implementaion.
Aim: optimizing the performance
AP_Presen-tation
SNR
PNR
Technical objects
2
3
1

Figure 16: Presentation process during data collection and updating

(1) No determination of presentation rules
Collected or updated data do not need predefined rules to assign a concrete portrayal number and position number in order to realize a performant presentation. The information for presentaion can directly be generated on the fly using the 3A data model as well as the ALKIS portrayal catalogue.

(2) Storing the presentation objects in the primary data base
During the collecting or updating process a presentation object is created to present a concrete portrayal, because the positioning of portrayals cannot be automatically generated for a specific map scale. In this case the information about the geometry, optianally a potrayal numer and position number are stored in the feature type.

(3) Determination of presentation rules
In order to minimize the runtime of a presentation a non spatial feature type AP_Presentation for a specific technical object can be generated during the data collecting or updating process. In this feature type a concrete portrayal or position number is stored, e.g. an arrangement of sysmbols in an area. The geometry for the presentation can be derived from the corresponding technical object on the fly using appropriate functions.

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 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 must also contain the attributes of the fundamental AFIS-ALKIS-ATKIS object, in order to be evaluated for the presentation in the presentation rules of the poortrayal catalogue.

So the thematic feature are instances of the same feature class and can carry identical attributs or associations as the DLM features that are refered by the association „is_derived_from“. But the model type code is differnt (e.g. DTK10, DTK25, DTK50, DTK100). By checking plausibility of the map geometry objects, it must be clear defined which attributes and associations are valid for these objects. Therefore there will be added model type codes in the UML model of the AdV or a clear mapping will be defined which model types include other model type codes (e.g. DLM 25 includes DTK 10 and DTK 25).

After the integration of the DTK in the AAA application schema (after version 5.1), DTK model types can be applied to map geometrie objects.

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”.

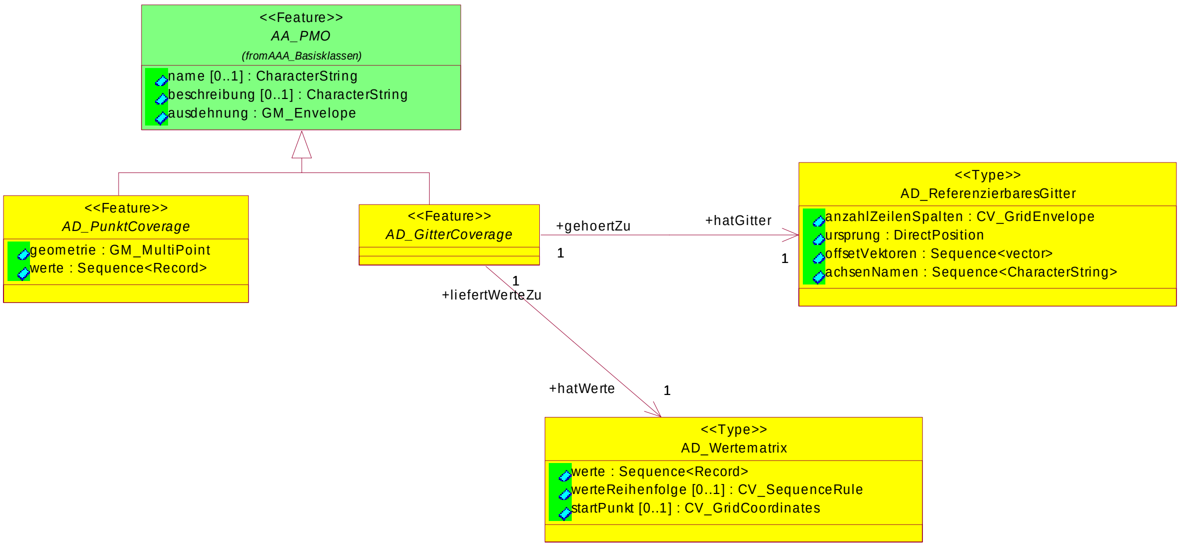

<<Feature>>
AA_PMO
(fromAAA_Basisklassen)
name [0..1] : CharacterString
beschreibung [0..1] : CharacterString
ausdehnung : GM_Envelope
<<Feature>>
AD_PunktCoverage
geometrie : GM_MultiPoint
werte : Sequence<Record>
<<Feature>>
AD_GitterCoverage
+gehoertZu
1
+hatGitter
1
<<Type>>
AD_ReferenzierbaresGitter
anzahlZeilenSpalten : CV_GridEnvelope
ursprung : DirectPosition
offsetVektoren : Sequence<vector>
achsenNamen : Sequence<CharacterString>
1
+liefertWerteZu
+hatWerte
1
<<Type>>
AD_Wertematrix
werte : Sequence<Record>
werteReihenfolge [0..1] : CV_SequenceRule
startPunkt [0..1] : CV_GridCoordinates

Figure 17: 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 Open Geospatial 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.

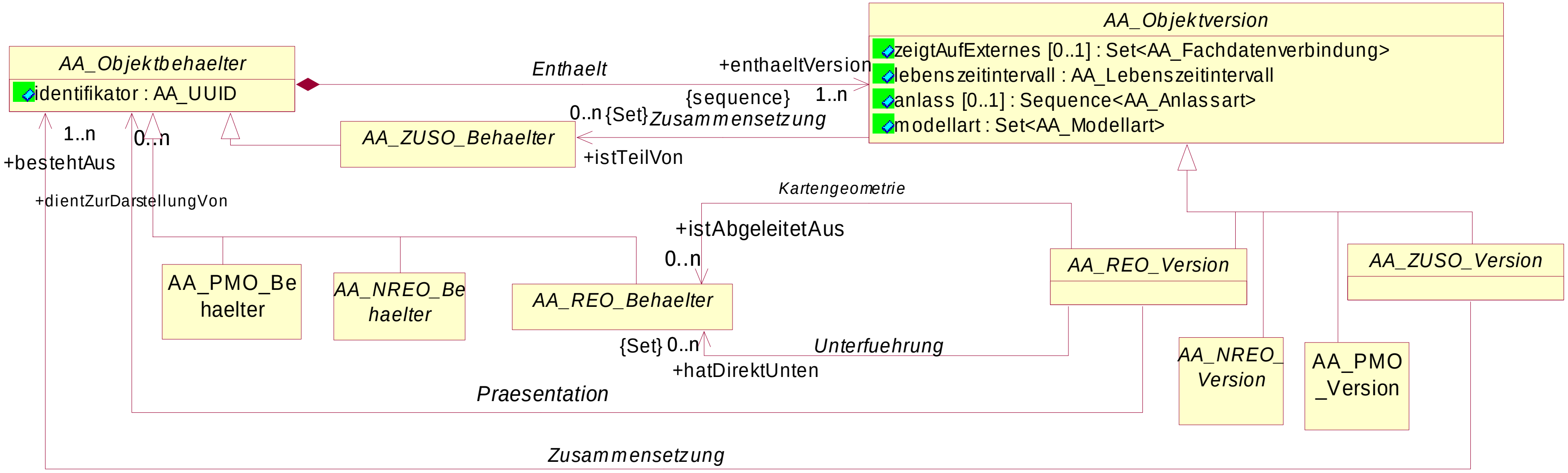


Figure 18: Versioning schema

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.

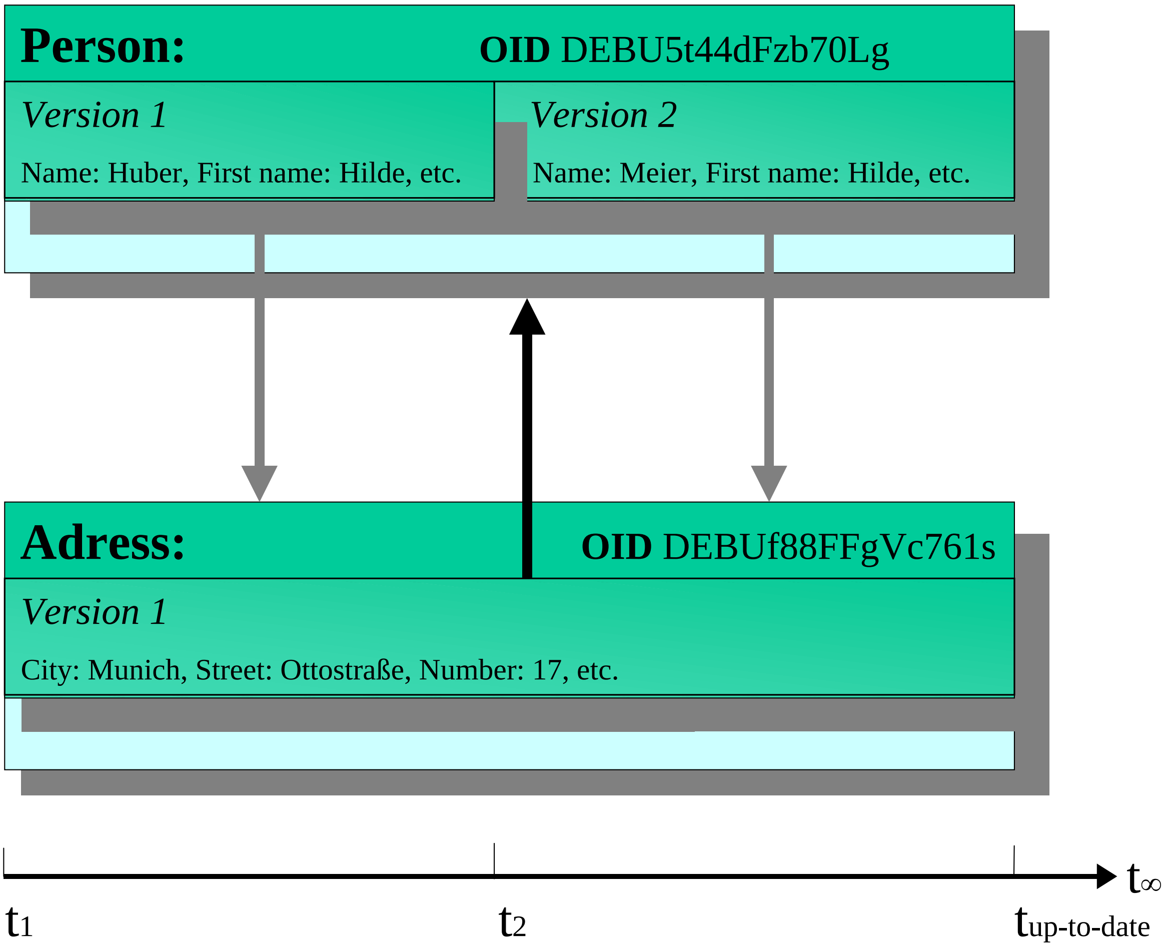

Person: OID DEBU5t44dFzb70Lg
Version 1
Name: Huber, First name: Hilde, etc.
Version 2
Name: Meier, First name: Hilde, etc.
Adress: OID DEBUf88FFgVc761s
Version 1
City: Munich, Street: Ottostraße, Number: 17, etc.
t∞
t1
t2
tup-to-date

Figure 19: 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.

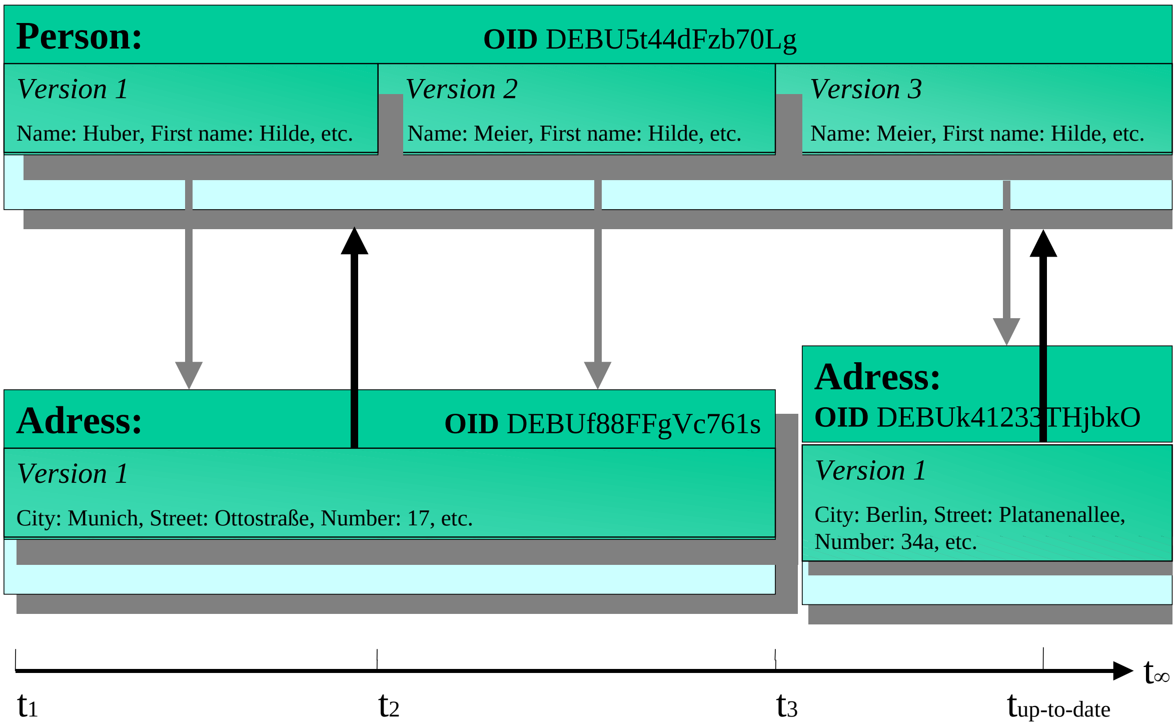

Person:
OID DEBU5t44dFzb70Lg
Version 1
Name: Huber, First name: Hilde, etc.
Version 2
Name: Meier, First name: Hilde, etc.
Version 3
Name: Meier, First name: Hilde, etc.
Adress:
OID DEBUf88FFgVc761s
Version 1
City: Munich, Street: Ottostraße, Number: 17, etc.
Adress:
OID DEBUk41233THjbkO
Version 1
City: Berlin, Street: Platanenallee, Number: 34a, etc.
t1
t2
t3
tup-to-date
t∞

Figure 20: 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.

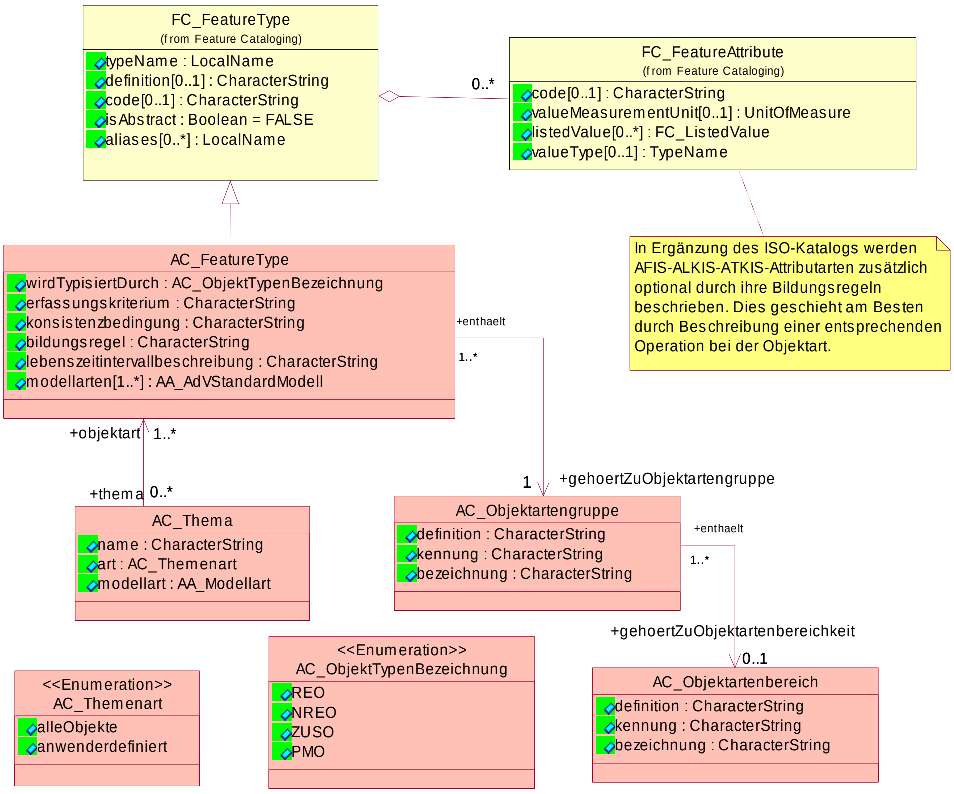

FC_FeatureType
(from Feature Cataloging)
typeName : LocalName
definition[0..1] : CharacterString
code[0..1] : CharacterString
isAbstract : Boolean = FALSE
aliases[0..*] : LocalName
0..*
FC_FeatureAttribute
(from Feature Cataloging)
code[0..1] : CharacterString
valueMeasurementUnit[0..1] : UnitOfMeasure
listedValue[0..*] : FC_ListedValue
valueType[0..1] : TypeName
In Ergänzung des ISO-Katalogs werden AFIS-ALKIS-ATKIS-Attributarten zusätzlich optional durch ihre Bildungsregeln beschrieben. Dies geschieht am Besten durch Beschreibung einer entsprechenden Operation bei der Objektart.
AC_FeatureType
wirdTypisiertDurch : AC_ObjektTypenBezeichnung
erfassungskriterium : CharacterString
konsistenzbedingung : CharacterString
bildungsregel : CharacterString
lebenszeitintervallbeschreibung : CharacterString
modellarten[1..*] : AA_AdVStandardModell
+enthaelt
1..*
+objektart
1..*
+thema
0..*
1
+gehoertZuObjektartengruppe
AC_Thema
name : CharacterString
art : AC_Themenart
modellart : AA_Modellart
AC_Objektartengruppe
definition : CharacterString
kennung : CharacterString
bezeichnung : CharacterString
+enthaelt
1..*
+gehoertZuObjektartenbereichkeit
0..1
<<Enumeration>>
AC_ObjektTypenBezeichnung
REO
NREO
ZUSO
PMO
<<Enumeration>>
AC_Themenart
alleObjekte
anwenderdefiniert
AC_Objektartenbereich
definition : CharacterString
kennung : CharacterString
bezeichnung : CharacterString

Figure 21: Expansion of standardised structure of the feature catalogue

3.7 Processes, operations and activities (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.

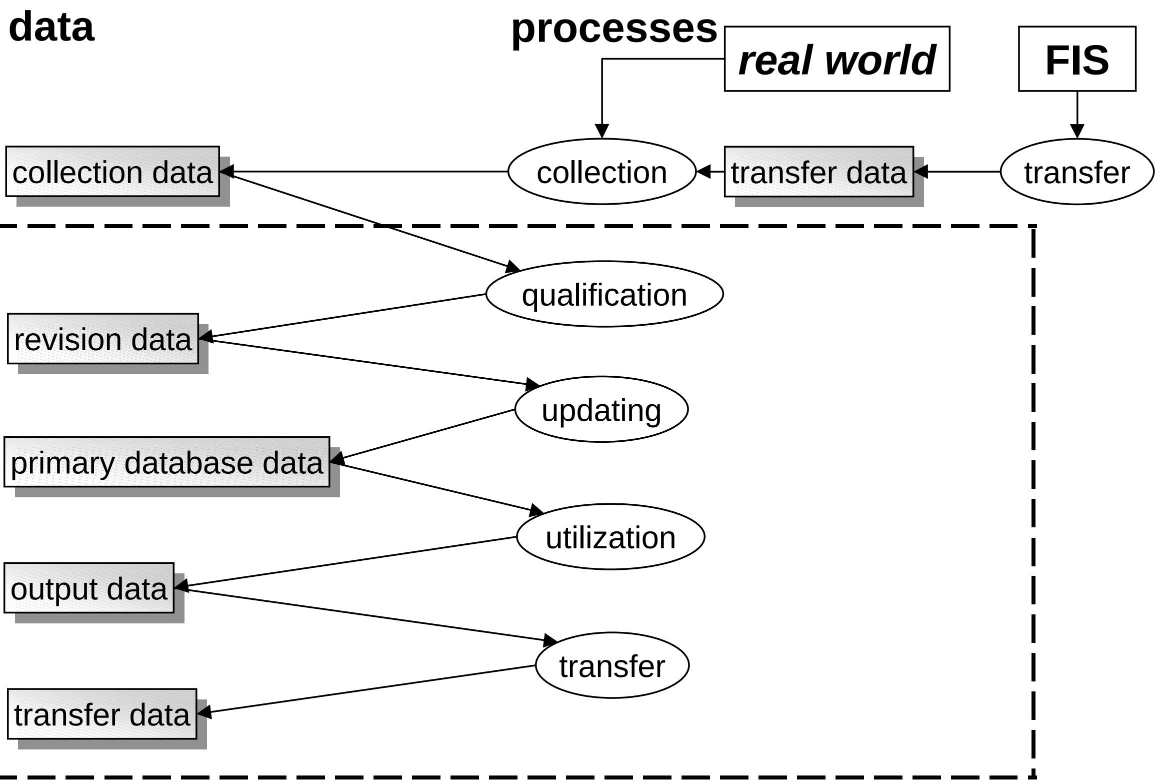


Figure 22: Processes and data for Geoinformation of Official Surveying and Mapping

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

- Activity 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 Operation and Activity

For a complete application description, operations and activities must be defined that set the data in functional dependencies and define the dynamic behaviour of the application. Operations belong to different processes of the AAA application schema. The following picture shows the depencencies between defferent processes.

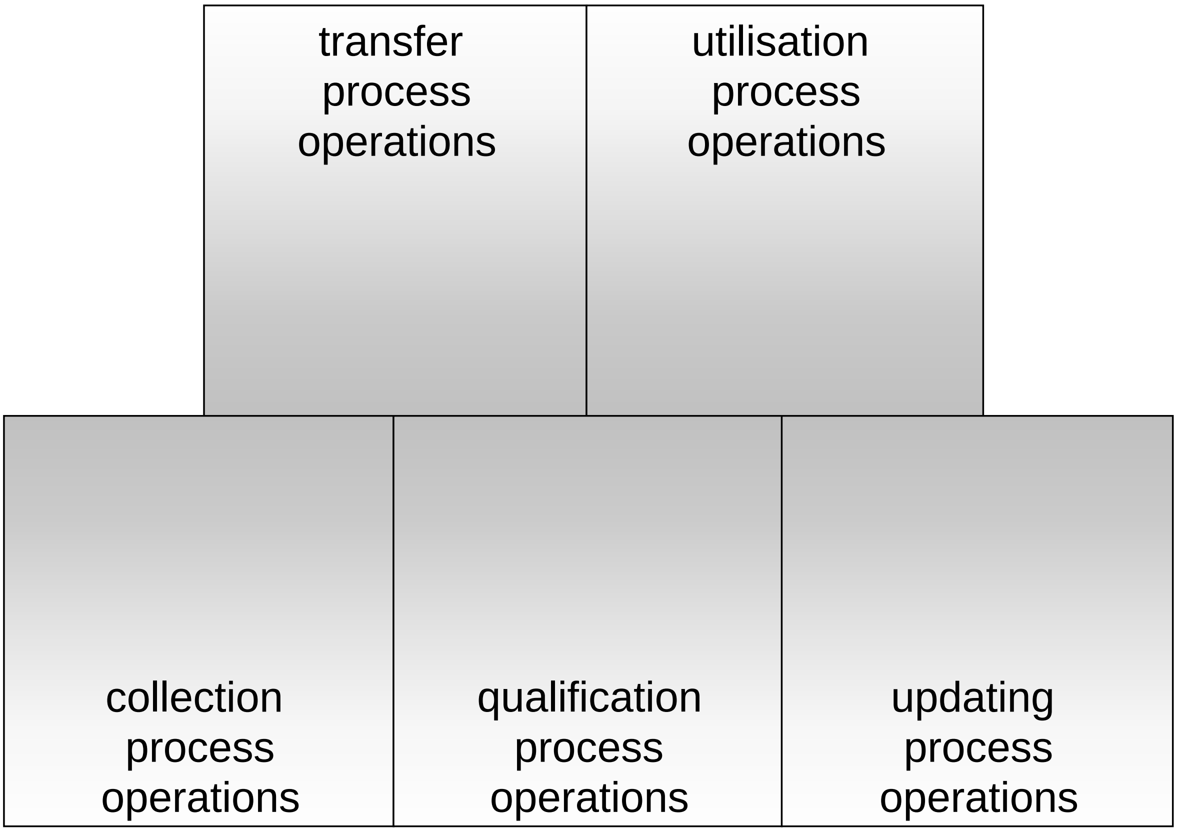

transfer process operations
utilisation process operations
collection process operations
qualification process operations
updating process operations

Figure 23: Operations in the AAA application schema

A operation contains the representation of processing steps for qualification, management, use and transfer, in which reference is made to various activities.

An activity 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 activities of other objects (input parameters). The result of the activity is prepared in the form of output parameters. Activities are defined in relation to the object and are part of a class within the UML model.

3.7.3 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.

3.7.3.1 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.

3.7.3.2 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.

3.7.3.3 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 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”.

3.7.3.4 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 are created (e.g. AX_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. 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.

3.7.3.5 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.

3.8 Project control in the Basic schema

The project control is integral part of the work- flow for updating the real estate cadastre. This work- flow can also be described as business process, which also can include the data collection in the field or the bill collection. The project control is only a part of such a business process and controls the work flow for updating the database in form of operations and activities. A connection to an external leaded business book will be realised. The updating of the real estate cadastre is structural standardized by implementation the project control. So the quality security and the law security for updating the real estate cadastre can be improved. Factual data of the feature classes can be tested for updating to defined times (production statement). The working steps can be visualized.

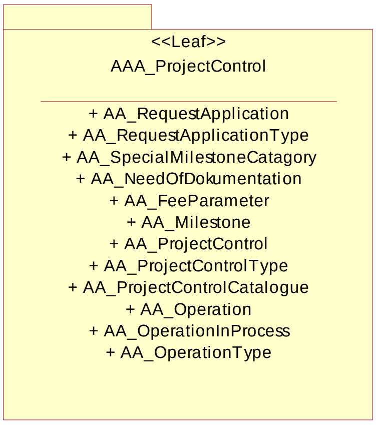


Figure 24: Package project control

Because of the general validity of the project administration in the AFIS-, ALKIS-, ATKIS application schema the project control is described in the basic schema as factual frame. The essential features of the project control can be used for other factual applications. The factual frame is specified by factual features of every federal state. The AAA- basic schema describes the project control for two different levels:

3.8.1 Meta model of the project control

The meta model consists of the principles of execution the processes (e. g. Sequence, Synchronization). A similar frame structure will be used by the modelling of the feature type catalogue in the basic application schema, whereby only the structure and not the technical content are described. The project control of this level is subdivided into request application type, which can be assigned to several project control types. An on the other hand several project control types can be assigned to fixed process types. The process types are described by concrete activities (function) as elementary component of a process. The connections and also the running of the processes will be documented as a rule in form of matrixes or connection catalogues.

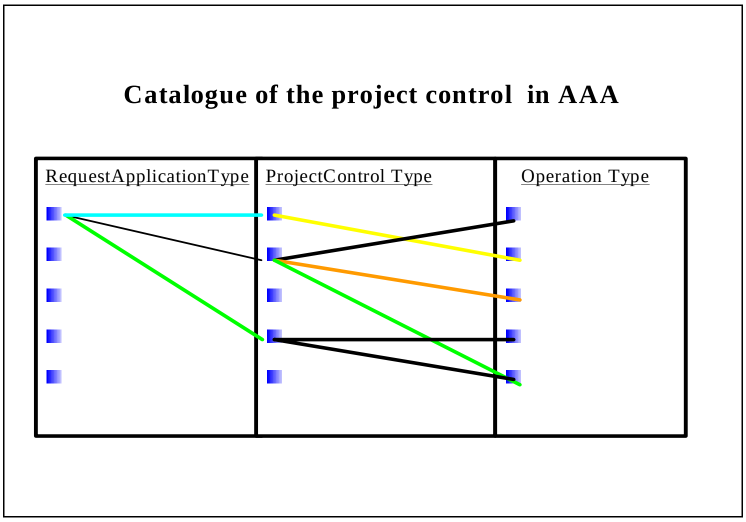


Figure 25: Request application – project control - operation

Feature Type AA_RequestApplicationType

The request application type defines a business process in the real estate cadastre. In order to realise the request application type several project control type are assigned. The request application type will be used for the assignment between updating request application and utilisation request application type to several project control types. In the real estate cadastre the following request application are possible to be used:

(1) Updating the real estate cadastre,

(2) Technical revision of real estate cadastre of the method of redistribution of parcels,

(3) Utilisation of the real estate cadastre.

Feature Type AA_ProjectControlType

The project control type combines project control feature types with the same characteristics. For a special request application type for example Updating the real estate cadastre, the project control type e.g. “FF- ALKIS” (Updating - ALKIS) will be chosen. This special project control type combines all updating causes und therefore all project control feature types for updating the real estate cadastre, whereby one project control feature type consists of several causes.

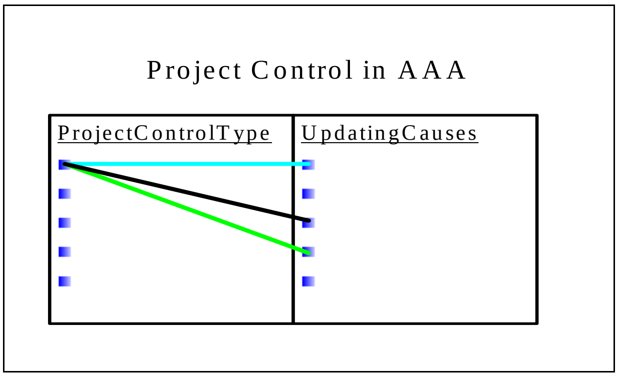


Figure 26: Project control – updating cause

In a project control database the special updating causes are assigned to updating the real estate cadastre, which are executed for a defined project control type. By selection the updating cause splitting or splitting parcels without measurements and marking the allowed combinations are also activated, so that they also can be executed in this process, so merging, changing by reason of the street law, changing by reason of water law, changing by reason of court decision, changing by reason of correction an error of measurement.

The codeliste of the update causes of the AdV of the GeoInfoDok 5.1 will be used as a technical frame. In context of the implementation in the federal states special feature types must be assigned to the special updating cause, e. g. splitting. These special feature types are allowed to be changed for this special updating cause, so e. g. the feature type AX_Parcel, AX_LocationDescriptionWithoutHouseNumber.

Also it is possible to define not allowed combinations of updating causes as technical necessity. For example it can be defined, that the updating of the owners data can only be executed in a separate project control type and not in the connection with the project control type “splitting”. This fact will be explained by the technical relation “IsNotCombinedWith” of the feature type “AA_ProjectControlType”.

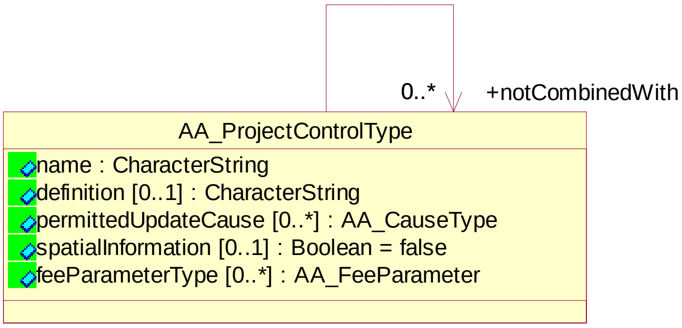

0..*
+notCombinedWith
AA_ProjectControlType
name : CharacterString
definition [0..1] : CharacterString
permittedUpdateCause [0..*] : AA_CauseType
spatialInformation [0..1] : Boolean = false
feeParameterType [0..*] : AA_FeeParameter

Figure 27: Relation “IsNotCombinedWith”

Feature type project control catalogue

The project control catalogue consists of the project control- and operationtype.

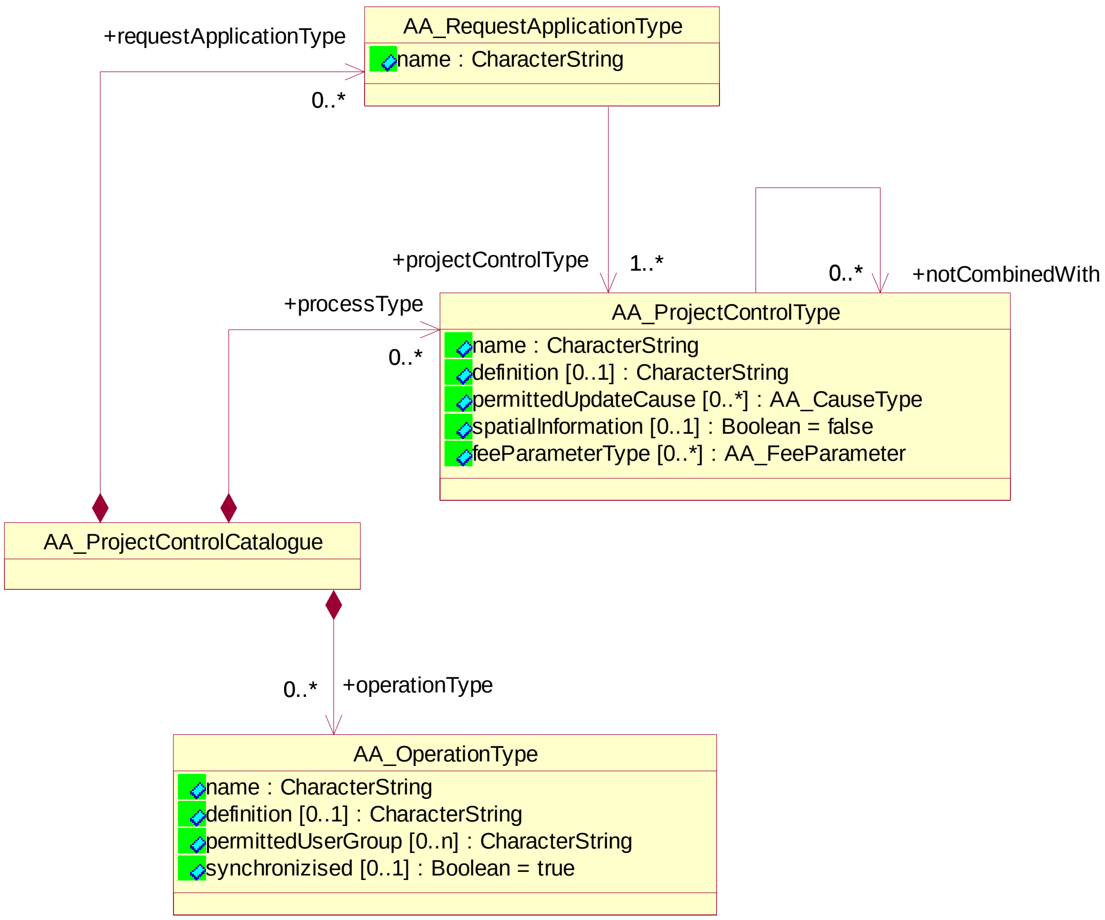

AA_RequestApplicationType
name : CharacterString
+requestApplicationType
0..*
+projectControlType
1..*
0..*
+notCombinedWith
+processType
0..*
AA_ProjectControlType
name : CharacterString
definition [0..1] : CharacterString
permittedUpdateCause [0..*] : AA_CauseType
spatialInformation [0..1] : Boolean = false
feeParameterType [0..*] : AA_FeeParameter
AA_ProjectControlCatalogue
0..*
+operationType
AA_OperationType
name : CharacterString
definition [0..1] : CharacterString
permittedUserGroup [0..n] : CharacterString
synchronizised [0..1] : Boolean = true

Figure 28: Project control catalogue

Feature type operation type

The feature type AA_OperationType combines several operations, which describe a common characteristic. For the updating cause “splitting or splitting parcels without measurement and marking” the following operations as operation types are defined for example:

(1) Preparatory works,

(2) Production statement (confirmation of the technical plausibility of statement marking, documentation the results of the survey, coordinates calculation, parcel area calculation, prerequisite feature creation,

(3) Completion and plausibility,

(4) Qualification,

(5) Updating decision.

Several operations can also be worked parallel (synchronization), in order to realize an optimal workflow. If an equal operation is used in parallel project control types and the operation is closed by the decision of the operator (milestone), this decision is transfered for all parallel working operations. This is only possible, if all operations have the same characteristic.

Such individual operation of a project control type can be assigned concrete activities. These activities can be assigned to an individual operation of a project control type or several operations. Inside of the operation “preparatory works” the following activities are defined:

(1) Preparing an output of data in primary database
Utilisation request is defined for preparing the data if need be preparing the cadastral field sheet,

(2) Carry out a Reservation (point code, parcel code, updating code),

(3) Implementation a project database,

(4) Giving up documents to survey institutions,

(5) Importing of collection data,

(6) Checking the Data of the owner and the register number,

(7) Checking the street code,

(8) Checking the parcel code and the point code,

(9) Checking the reservation of the point code,

(10) Checking the double number,

(11) Creating temporal documents on conveyance of real estate,

(12) ……

The relationship of assignment between the activities and the operation can be shown from the figure above.

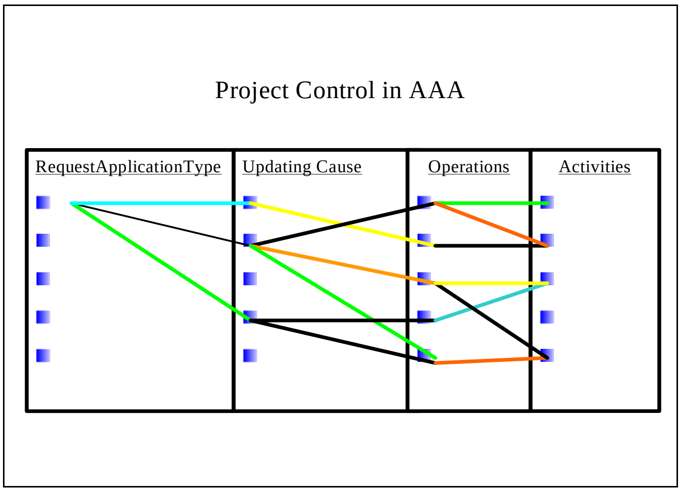


Figure 29: Request Application - operation – activities

The existence of the production statement of the created survey documents (Document: cadastral field sheet, statement of marking) will be documented by the operation “production statement”.

In the operation “Completion and plausibility” the object structured collection data are taken over in the project database. In the following step the collection data will be completed to ALKIS- concurring features with the following solution of plausibility.

Inside of the operation the following cause types are worked

(1) Updating Cause “Subdivision or splitting parcels without measurements and marking”

(2) Updating Cause “ Updating other data”

(3) Updating cause “Updating Land register”.

In the operation “Qualification” the whole project database will be tested and the result is documented in a protocol of qualification. The correction of the mistakes is made in the operation “Completion and plausibility”. Functions of qualification are: checking the causes, checking the actuality, checking identical point code, checking several calculated point location.

In the operation “updating decision” the available updating request will be set free for the updating the database.

3.8.2 Modelling of the project control

The feature types are only modelled, which managed the project control in the AAA- database on the level of instances of the feature types. These are corresponding the structure of the meta model: AA_Application, AA_ProjectControl, AA_Operation and AA_Activity. In this process special parameter are defined and analysed (e. g. milestone), which document and lead the concrete workflow.

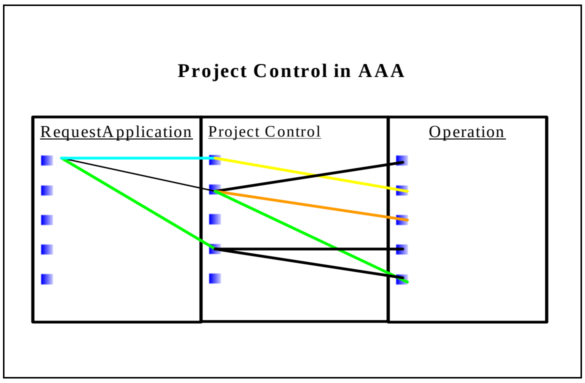


Figure 30: Project control – instances

In the following document the mechanism of the project control for updating the database is described (figure 30) in connection to the example of the splitting survey.

Feature type AA_RequestApplication

This feature type realised a “mini administration”, that means an interface to the external request application administration (Business book). So it is possible by an entry in the external request application administration to have a direct connection to this request application in an technical schema (e. g. spatial).

The request application feature type managed also the renewed submission of the request application and supports the controlling of the project control feature type. With the data of the spatial reference it is possible to search after existing processes, in order to determine competed applications or in order to consider other neighbouring applications by the working. The technical order competed request applications is to be determined by the administrator.

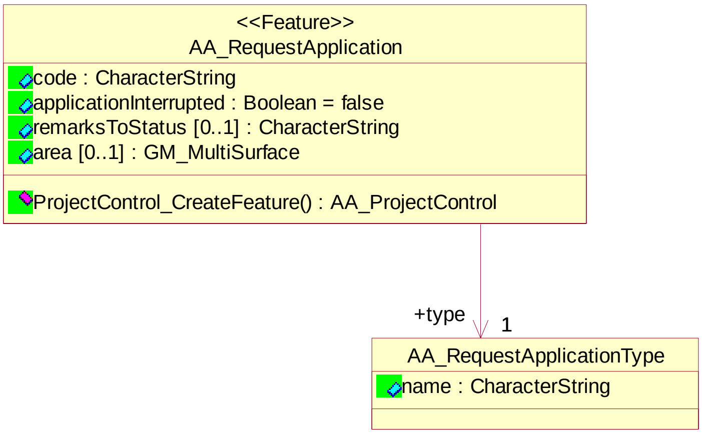

<<Feature>>
AA_RequestApplication
code : CharacterString
applicationInterrupted : Boolean = false
remarksToStatus [0..1] : CharacterString
area [0..1] : GM_MultiSurface
ProjectControl_CreateFeature() : AA_ProjectControl
+type
1
AA_RequestApplicationType
name : CharacterString

Figure 31: AA_RequestApplication

Feature type AA_ProjectControl

The application feature type is combined with the project control feature type (AA_ProjectControl), in order to fix the assignment of an application to one or more project control feature types and in order to control the not allowed combinations. Furthermore the project control feature type manages und controls the concrete workflow of the operations, e. g. in the part of the process “technical qualification”. The update causes and also the causes of utilisation are leaded by the project control feature type.

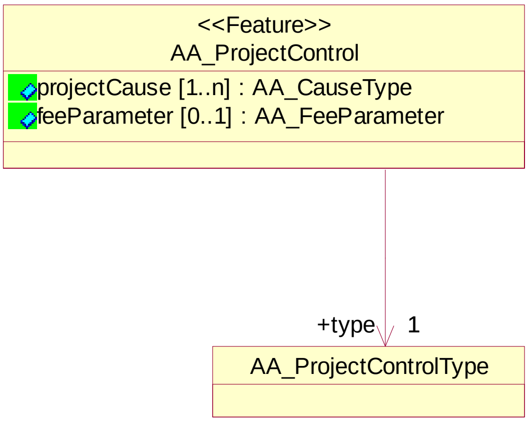

<<Feature>>
AA_ProjectControl
projectCause [1..n] : AA_CauseType
feeParameter [0..1] : AA_FeeParameter
+type
1
AA_ProjectControlType

Figure 32: AA_ProjectControl

Feature type AA_Operation

The operation is part of the project control and consists of several work steps. The operations are defined as closed work steps. A defined work flow fixed the order und the dependence of the operations and the work steps. The operations are combined into groups and they are worked in a special order serial or parallel. The decision of the end of an operation is documented by the status (milestone).

The general model of operation of projects distinguishes the following types:

- Project with one or more updating causes with the duty of documentation
- Project with one or more updating causes without the duty of documentation

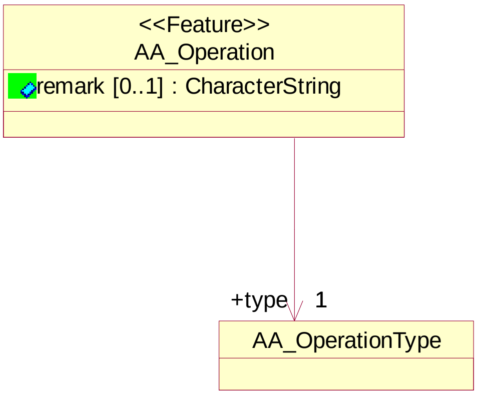


Figure 33: AA_Operation

In the operation the data type AA_Milestone will be occupied for registration the actual status and the responsibility of the operation working with the data of the beginning, finishing, comment etc. This can be seen in the following figure. With a relation the connection is assigned to the defined user profile in the project working, in order to legitimate the right to a concrete operation working.

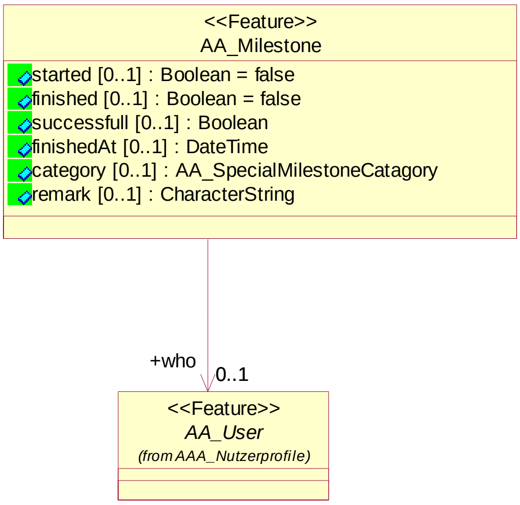

<<Feature>>
AA_Milestone
started [0..1] : Boolean = false
finished [0..1] : Boolean = false
successfull [0..1] : Boolean
finishedAt [0..1] : DateTime
category [0..1] : AA_SpecialMilestoneCatagory
remark [0..1] : CharacterString
+who
0..1
<<Feature>>
AA_User
(from AAA_Nutzerprofile)

Figure 34: AA_Milestone

A technical program must support the function, that the milestone is only set, if all the necessary activities are executed successfully.

The activities are assigned to the operations, which contains the functions to execute the process. It must to be distinguished, if the activities are necessary for the working, shows a help or is not allowed to be used. This will be managed by using a concrete value of the following codeliste.

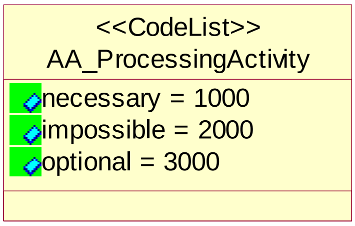


Figure 35: Status “Activities”

Feature Type “request application area”

The request applications can be stored and updated as data in primary database in a local update unit (local data management). By using the NBA-method in the AAA_Programm an updating can be managed of the data in primaray database at least daily. The request applications are also managed with the NBA- method. The advantage of this process is, that the request application area can be updated alone and passed on with NAS

With an information and presentation component of the data in primary database it is possible to present request applications. The spatial reference can be expressed by area. In order to get Information of the request application area, the request application code will be presented at the standard position of the area.

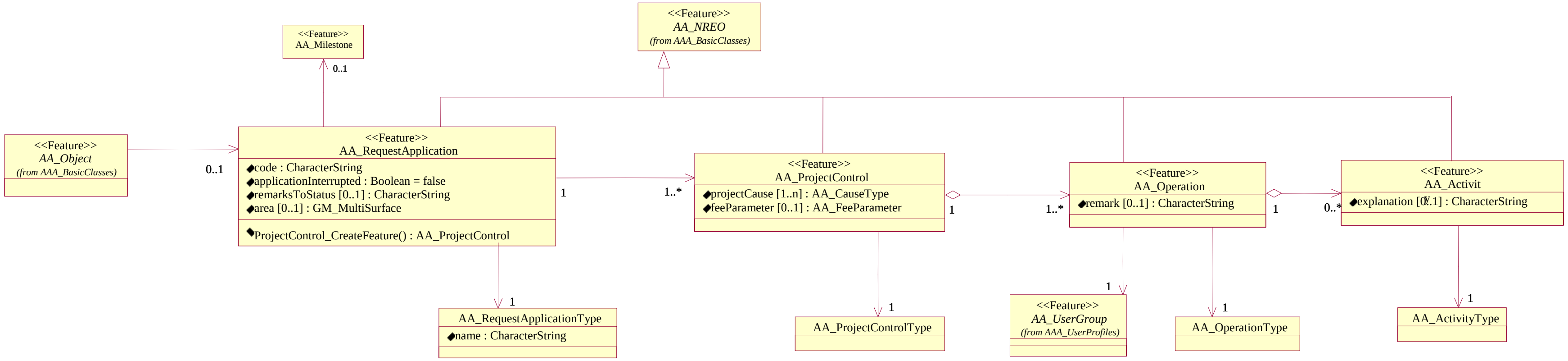


Figure 36: Overview of the project Control

3.9 Principles of using filter encoding

To code a selection, the <wfs:Query> element from the “Web Feature Service“ specification (WFS) of the Open GIS Consortiums in Version 1.0.0 is used in the NAS. Several queries may occur in one selection, whereby each query relates to an instance-capable feature type. The various queries have a complementary effect.

The current WFS specification supports only the data of the concrete, instance-capable feature types, i.e. the inheritance hierarchy modelled in the AAA application schema is not supported. It is therefore not possible, for example, to send one single query for “AX_actual use“, in order to request all TN objects, instead a <wfs:Query> element must be indicated for each feature type. An upgrade for supporting the inheritance hierarchy would be possible in principle, although only AdV-specific.

A <wfs:Query> includes an embedded <ogc:Filter> element for filtering the objects from the entire scope of the feature type. An <ogc:Filter> output consists of a predicat, which is applied for each object of the feature type in the database on which the search is to be carried out. An object becomes part of the selection only if it fulfils the predicate. The predicates are to be regarded such that they exert a fundamental effect on the XML instances, which represent these objects.

Accordingly, the predicate consists of a Boolean expression that can have any number of atomic operators that are linked via

- the logical operators
    - <ogc:And>
    - <ogc:Or>
    - <ogc:Not>

.

The following are supported for the atomic operators

- spatial operators
    - <ogc:Equals>
    - <ogc:Disjoint>
    - <ogc:Touches>
    - <ogc:Within>
    - <ogc:Overlaps>

  - <ogc:Crosses>
  - <ogc:Intersects>
  - <ogc:Contains>
  - <ogc:DWithin>
  - <ogc:Beyond>
  - <ogc:BBOX>

and

- comparative operators
  - <ogc:PropertyIsEqualTo> (=)
  - <ogc:PropertyIsNotEqualTo> (<>)
  - <ogc:PropertyIsLessThan> (<)
  - <ogc:PropertyIsGreaterThan> (>)
  - <ogc:PropertyIsLessThanOrEqualTo> (=<)
  - <ogc:PropertyIsGreaterThanOrEqualTo> (>=)
  - <ogc:PropertyIsLike> (text comparison with „Wildcards“ for one or several symbols)
  - <ogc:PropertyIsNull> (check for missing value)
  - <ogc:PropertyIsBetween> (combination of >= and <=)

. The meaning of the logical operators and the comparative operators is derived from the meaning used in the SQL and/or expressed directly with the name.

The meaning of the spatial operators is generally defined in the OpenGIS Simple Features specification and included in the Filter Encoding. <ogc:Intersects> is presumably the most important operator that reports “true“ if two geometries are not non-intersecting. <ogc:BBOX> is a simplified form that permits as test geometry only one Bounding Box. <ogc:Disjoint> is the inverse of <ogc:Intersects>. <ogc:Contains> or <ogc:Within> are to be used in cases of genuine “contain” rather than overlap. For further questions, see OpenGIS specification Filter Encoding and Simple Features for SQL.

For spatial operators and comparative operators, a property of the object for which the comparison is to be carried out is generally indicated.

This takes place using Xpath, although we recommend limiting to the short form. This means:

- An attribute “att” of query feature type is referenced as follows:

```
<ogc:PropertyName>att</ogc:PropertyName>
```

or with a definitive example from the AAA application schema:

```
<ogc:PropertyName>parcel designation</ogc:PropertyName>
```

- If “att“ is an attribute of the query feature type and the value of the attribute is of type “AX_DT” and attribute “att2” is to be referenced, this takes place as follows:

  ```
  <ogc:PropertyName>att/AX_DT/att2</ogc:PropertyName>
  ```

  or with a definitive example from the AAA application schema:

  ```
  <ogc:PropertyName>lifetime interval/AA_lifetime interval/ends</ogc:PropertyName>
  ```

- A relation (to be more precise the role in the definition direction of a relation) “rel” of the query feature type is referenced as follows:

  ```
  <ogc:PropertyName>rel</ogc:PropertyName>
  ```

  If this is feature type “AX_OA” as a relation partner and has an attribute “att3”, it is referenced as follows:

  ```
  <ogc:PropertyName>rel/AX_OA/att3</ogc:PropertyName>
  ```

  or with a definitive example from the AAA application schema (via two relations):

  ```
  <ogc:PropertyName>
          is posted/AX_posting location/to/AX_posting location/continuous number
  </ogc:PropertyName>
  ```

- In the event that an XML attribute has to be definitively referenced and evaluated (e.g. “xlink:href”, “uom” or “srsName”), this takes place as follows:

  ```
  <ogc:PropertyName>att/@xmlatt</ogc:PropertyName>
  ```

  or with a two definitive example from the AAA application schema:

  ```
  <ogc:PropertyName>official area/@uom</ogc:PropertyName>
  <ogc:PropertyName>is posted/@xlink:href</ogc:PropertyName>
  ```

It is assumed that the default namespace of the XML documents is "http://www.adv-online.de/namespaces/adv". Otherwise, all identifiers are to be qualified by the namespace code (as has already taken place in the example of xlink:href above).

```
<ogc:PropertyName>adv:att</ogc:PropertyName>
```

instead of

```
<ogc:PropertyName>att</ogc:PropertyName>
```

In the case of simple attributes, the comparative operator is generally compared to the attribute value with a fixed value (element <ogc:Literal>), e.g.

```
<ogc:PropertyIsEqualTo>
        <ogc:PropertyName>location type</ogc:PropertyName>
        <ogc:Literal>1100</ogc:Literal>
</ogc:PropertyIsEqualTo>
```

which is satisfied for all objects in the database, for which the location type attribute exhibits a corresponding value (value type 1100) or

```
        <ogc:PropertyIsGreaterThanOrEqualTo>
        <ogc:PropertyName>
                        lifetime interval/AA_lifetime interval/begins
                </ogc:PropertyName>
                <ogc:Literal>2003-05-20T00:00:00+02:00</ogc:Literal>
        </ogc:PropertyIsGreaterThanOrEqualTo>
```

or

```
        <ogc:PropertyIsLessThan>
                <ogc:PropertyName>
                        lifetime interval/AA_lifetime interval/ends
                </ogc:PropertyName>
                <ogc:Literal>2003-05-20T00:00:00+02:00</ogc:Literal>
        </ogc:PropertyIsGreaterThanOrEqualTo>
```

or in the event of a comparison to the value NULL

```
        <ogc:PropertyIsNull>
                <ogc:PropertyName>
                        lifetime interval/AA_lifetime interval/ends
                </ogc:PropertyName>
        </ogc:PropertyIsNull>
```

The comparison below would be used to check for values in an area, e.g. whether the location type is a value in the 1xxx range:

```
        <ogc:PropertyIsBetween>
                <ogc:PropertyName>location type</ogc:PropertyName>
                <ogc:LowerBoundary>
                        <ogc:Literal>1000</ogc:Literal>
                </ogc:LowerBoundary>
                <ogc:UpperBoundary>
                        <ogc:Literal>1999</ogc:Literal>
                </ogc:UpperBoundary>
        </ogc:PropertyIsBetween>
```

Equally a predicate for parcels with an official area of at least 1000 m^2 and no more than 2000 m^2:

```
        <ogc:PropertyIsBetween>
                <ogc:PropertyName>official area>/ogc:PropertyName>
                <ogc:LowerBoundary>
                        <ogc:Literal>1000</ogc:Literal>
                </ogc:LowerBoundary>
                <ogc:UpperBoundary>
                        <ogc:Literal>2000</ogc:Literal>
                </ogc:UpperBoundary>
        </ogc:PropertyIsBetween>
```

The LIKE comparison is helpful for flexible text comparisons. The following predicate filters out all addresses whose telephone number starts with 0228

```
<ogc:PropertyIsLike wildCard="*" singleChar="?" escape="\">
    <ogc:PropertyName>telefon</ogc:PropertyName>
    <ogc:Literal>0228*</ogc:Literal>
</ogc:PropertyIsLike>
```

while the predicate below filters those persons for whom a birth name is set, begins with a “M” and has a “t” as the third and fourth letters:

```
<ogc:PropertyIsLike wildCard="*" singleChar="?" escape="\">
    <ogc:PropertyName>geburtsname</ogc:PropertyName>
    <ogc:Literal>M?tt*</ogc:Literal>
</ogc:PropertyIsLike>
```

In the case of spatial operators, a comparison is made between a property (the name of the geometric attribute type) and a fixed geometry in the same manner as the comparisons between a textual or numerical property and a fixed value. In the case of spatial operators, the fixed value is always expressed by a <ogc:Literal> element through the respective GML geometric element, for example

```
<ogc:Intersects>
    <ogc:PropertyName>generator</ogc:PropertyName>
    <gml:Polygon>
    <gml:exterior>
            <gml:Ring>
                <!—outline of the search area here -->
            </gml:Ring>
        </gml:exterior>
    </gml:Polygon>
</ogc:Intersects>
```

If the overall key for a catalogue entry is known, the corresponding catalogue entry may, for example be requested with a query of the following type (in this case the district with code “071234“)[fn1]:

```
<wfs:Query typeName="AX_district">
    <ogc:Filter>
        <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>key overall</ogc:PropertyName>
            <ogc:Literal>071234</ogc:Literal>
        </ogc:PropertyIsEqualTo>
    </ogc:Filter>
```

Instead of

```
$1 = concat (from posting sheet district/state, from posting sheet district/district)
```

the correct designation is, for example

```
$1 = concat (from posting sheet district/AX_posting sheet district/state,
            from posting sheet district/AX_posting sheet district/district)
```

```
	</wfs:Query>
```

If all catalogue entries with a certain sub-key are to be requested, either <ogc:PropertyIsLike> or a comparative operator can be used to search for the individual attributes of the key data type. All districts in the state are found, for example, with:

```
	<wfs:Query typeName="AX_district">
		<ogc:Filter>
			<ogc:PropertyIsLike wildCard="*" singleChar="?" escape="\">
				<ogc:PropertyName>key overall</ogc:PropertyName>
				<ogc:Literal>07*</ogc:Literal>
			</ogc:PropertyIsLike>
		</ogc:Filter>
	</wfs:Query>
```

or

```
	<wfs:Query typeName="AX_district">
		<ogc:Filter>
			<ogc:PropertyIsEqualTo>
				<ogc:PropertyName>
					key/AX_district_key/state
				</ogc:PropertyName>
				<ogc:Literal>07</ogc:Literal>
			</ogc:PropertyIsEqualTo>
		</ogc:Filter>
	</wfs:Query>
```

In addition to the filter condition, further elements can be embedded into the <wfs:Query>-element. The two elements <wfs:XlinkPropertyName> and <wfs:XlinkPropertyPath> can be used to incorporate in one action further objects into the groups of results using a query. This procedure can be employed to significantly reduce the number of queries – and therefore also utilisation jobs.

The element

```
	<wfs:XlinkPropertyName traversalXlinkDepth="1">
		belongs proportionately to
	</wfs:XlinkPropertyName>
```

in an AX_parcel query results in all relation partners up to a depth of 1 (i.e. the direct relation partners, in this case the affected parcel) along the belongs proportionately to relation are incorporated into the group of results.

A refining of this is <wfs:XlinkPropertyPath>, which results in the object itself along the path rather than the width being incorporated into the group of results:

```
	<wfs:XlinkPropertyPath>
		is posted/AX_posting location/is part of/AX_posting location
	</wfs:XlinkPropertyPath>
```

If only individual, quite specific subordinate objects are required (in the example only for a small number of parcels in the posting sheet), the selection can usually be split into two queries. The first query for selecting the parcel and then the selection of the posting sheets.

In the NAS, all relations are presented in only one direction marked as navigable in the UML model. The query below requests all parcels and the locations to which they are posted:

```
    <wfs:Query typeName="AX_parcel">
    <wfs:XlinkPropertyPath>is posted/AX_posting location</wfs:XlinkPropertyPath>
    </wfs:Query>
```

Or in the event that the parcel is known, the identifier for the posting location can be extracted from the <is posted> (the string after the “urn:adv:” prefix in the example “DEBY123412345678”) and the posting location can be requested as follows:

```
    <wfs:Query typeName="AX_posting location">
        <ogc:Filter>
            <ogc:FeatureId fid="DEBY123412345678"/>
        </ogc:Filter>
    </wfs:Query>
```

In the reverse direction, i.e. from the posting location to the parcel, the relation is in fact also named (“plot consists of”) although not represented in the NAS. If the parcels that are assigned via “is posted” to a specific posting location (the ID “DEBY123412345678” is used again in the example) are now to be determined, this can take place through checking the relation:

```
    <wfs:Query typeName="AX_parcel">
        <ogc:Filter>
        <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>is
posted/@xlink:href</ogc:PropertyName>
            <ogc:Literal>urn:adv:DEBY123412345678</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        </ogc:Filter>
    </wfs:Query>
```

An equivalent query (insofar as parcels and posting locations are present in the same local database)

```
    <wfs:Query typeName="AX_parcel">
        <ogc:Filter>
        <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>
                    istGebucht/AX_posting location/@gml:id
                </ogc:PropertyName>
            <ogc:Literal>DEBY123412345678</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        </ogc:Filter>
        </wfs:Query>
```

There is no option such as “XlinkPropertyPath” also in an inverse direction, i.e. the simultaneous selection of certain posting locations and all parcels posted to them. The selection must always be made in two steps, i.e. via two queries.

Several queries are usually required to obtain from the database objects that are needed for more complex queries. The new query is formulated from the results of the previous query. Accesses to catalogue entries for decrypting key values are frequently required.

References:

- OGC Implementation Specification Filter Encoding 1.0.0 (http://www.opengis.org/techno/specs/02-059.pdf)
- OGC Implementation Specification Web Feature Service 1.0.0 (http://www.opengis.org/techno/specs/02-058.pdf)
- OGC Implementation Specification Simple Features for SQL 1.1 (http://www.opengis.org/techno/specs/99-049.pdf)
- GeoInfoDok 5.1 – Section 7.2 Processes in ALKIS – Part B (http://www.adv-online.de/veroeffentlichungen/afis-alkis-atkis/Dokumente/Prozesse in ALKIS.pdf)
- GeoInfoDok 5.1 – Section 10
- Xpath (http://www.w3.org/TR/xpath)

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 corresponding ZIP archive.

5.2 Explanations on AFIS

This section contains the technical explanations for the AFIS-application. This section is available as a separate PDF-file.

5.3 Explanations on ALKIS

In this chapter, the structure and content of the ALKIS-application are explained. The structure of this document is orientated at the structure of the ALKIS feature catalogue. Almost all object types (features) are commented and examples are shown. The explanations for the ALKIS-application are available as a separate 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 separate PDF-file.

5.5 Explanations on ATKIS Digital Terrain Models

In this chapter, the structure and content of the ATKIS Digital Terrain Model-application are explained. For special feature types detailed explanation and examples are provided.

The explanations for the ATKIS Digital Terrain Model-application are available as a separate PDF-file.

6 AFIS catalogue documents

6.1 AFIS feature catalogue

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

6.2 Processes in AFIS

The “Processes in AFIS“ document contains two parts:

- The output catalogue that is derived from the Rational-Rose-model. This catalogue is available as a separate PDF-File and an HTML-Document.
- The Filter Encoding expressions for the AFIS standard outputs as a separate document.

6.3 AFIS portrayal catalogue

The AFIS portrayal catalogue will be created in accordance to the ALKIS portrayal catalogue.

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.
  Both sections are available as separate PDF document or HTML document. The HTML document do not have preliminary remarks.

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 consists of the following parts:

- The output catalogue that ist derived from the Rational-Rose-model. This catalogue is available as a separate PDF-File.
- The Filter Encoding expressions for the ALKIS standard outputs (XSLT scripts) as a separate document.

All parts are collected in one ZIP 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
- examples
- positioning rules

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.
  Both sections are available as separate PDF document or HTML document. The HTML document do not have preliminary remarks.

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 19115:2002; Metadata catalogue - 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 within 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 *Open Geospatial Consortium* (OGC), for encoding geoinformation, which is also provided for standardization in the ISO 19100 series as ISO 19136. GML is – simplified said – a framework for modelling application specific feature types including their properties in XML schema. For describing feature types with their numeric, txtual, geometrical, temporal and other properties numerous of often used standardised components can be used, e.g. geometry types. This way a application specific XML language is defined in a GML application schema. GML is gradually achieving market acceptance.

Since the NAS contains besides the encoding of technical objects also operation modelled in the application schema in order to maintain data in primary database (update, initialisation, lock/unlock of objects, reservation of technical codes, request of output products including User-specific updating of secondary database). Therefore the GML feature types are embedded within operations that are principally web services, putting into account specific elements of the OGC specifications Web Feature Service (WFS) and Filter Encoding (FES) that are complementary to GML. In this sense, a AAA-compliant database can be compared with an embedded Web Feature Server that aditionally considers AAA-specific requirements.

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 operations for updating and request of GML-data , however, this is currently only possible to a certain degree and by using AFIS-ALKIS-ATKIS-specific extensions.

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 technical- and basic model on a GML-3.0 application schema that is specified in the NAS by schema components (including additional corresponding schema files, see clause 10.1.3.4) of the schema file AAA technical schema.xsd (de: AAA-Fachschema.xsd).
  The conceptual schema of the NAS-operations is mapped in XML schema components by using the “NAS Encoding Rules” and WFS/FES-XML-schemas. It is specified in the schema file NAS-operations.xsd (de: NAS-Operationen.xsd).
- 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". This means that the NAS uses GML in compliance with the requirements of the GML specification.
- 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 Geospatial Consortium, 2003
  Online
- Web Feature Service (WFS) 1.0
  Open Geospatial Consortium, 2002
  Online
- Filter Encoding (FES) 1.0
  Open Geospatial Consortium, 2002
  Online
- OWS Common Implementation Specification 1.0
  Open Geospatial Consortium, 2005
  Online

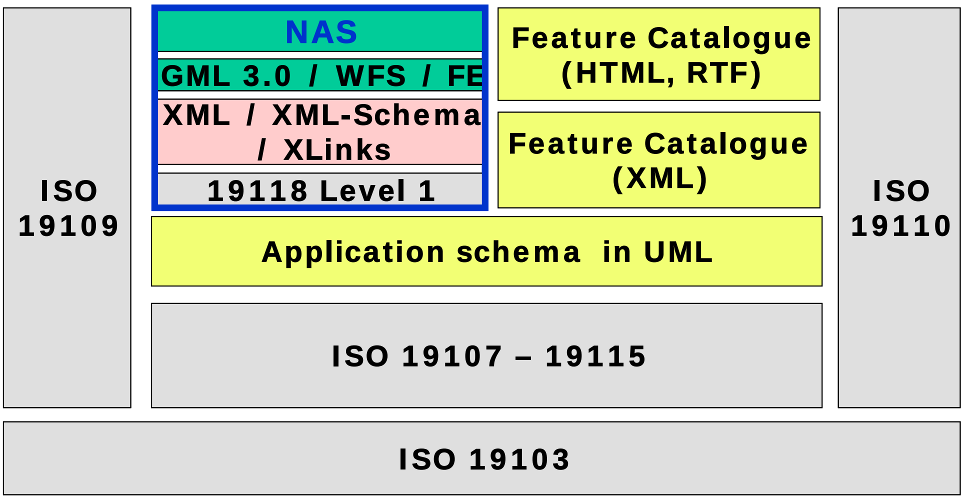


Figure 37: Embedding the NAS in de-jure standards and de-facto standards

Since GML application schemas use standardised XML components, e.g. for geometry types, and there are rules in GML for using XML schema in order to define a application schema, it is generally possible that generic GML software – that has been implementated the applied XML components - is able to conduct and syntactically interpretate AFIS-ALKIS-ATKIS-objects by analysing the GML application schema of the NAS. This is possible even if the software knows nothing about the NAS and AFIS-ALKIS-ATKIS in advance.

In the definition of the NAS GML profile (see chapter 10.1.6) For that reason additional requirements are specified and documentated for the capabilities of GML software. With determination of this profile it was considered in the scope that this profile meets also requirements of applications beyond the AFIS. ALKIS and ATKIS concept in order to come from an AdV internal specification to a better acceptance in a broader context.

By defining the NAS specification as operations and not as pure „data exchange format“, the GML features in the NAS are embedded in most cases within the XML elements of operation requests and responses. For example of the output of the primary data base, the list of GML features (the GML document) is embedded in the NAS response document and therefore it can be realised and extracted in an easy way.

10.1.2 Encoding process

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

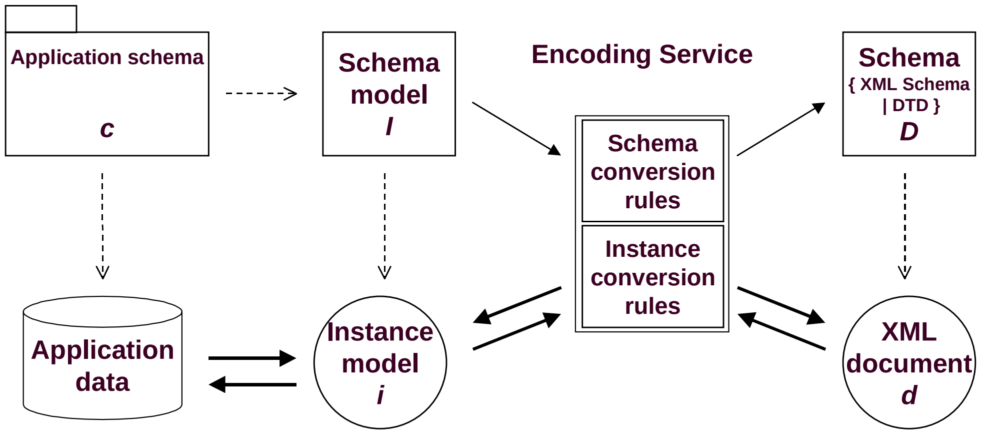


Figure 38: XML-based encoding regulations in accordance with ISO 19118

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:

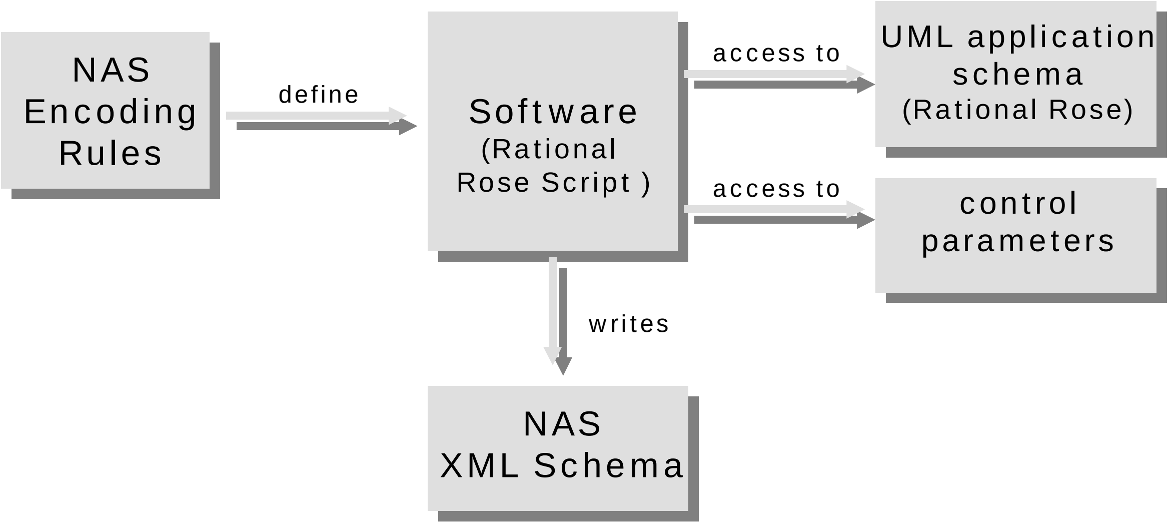


Figure 39: Automatic derivation of the NAS schemas

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>>.

Packages with the *Stereotype* <<Application Schema>> are qualified by control opaerations. The control parameter “targetNamespace” describes the XML namespace of the application schema, “xmlns” stands for the short term that has to be used within the XML schema documents; both control parameters must be applied at the packages using the stereotype. The control parameter “ISO19100” must be “true” if the content of the packages are parts of the conceptual model of the standard serie of ISO 19100. If the name of the corresponding XML schema file shold be different of the name of package, the name of the file can be applied in the control parameter “file name” (without the extension “xsd”).

Additionally, all dependencies between packages with *Stereotype* <<Application Schema>> must be completely activated.

The use of inheritance is permitted only for objects with identity (*Stereotypes* <<Feature>>, <<Type>> and without *Stereotype*) and for enumerations that are completely defined in the model (*Stereotypes* <<Enumeration>> and <<CodeList>>). 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", “ImplementedByType”, “ExtendsTypes”, "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
<<Application Schema>> package	XML schema file

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

Packages with Stereotype <<Application schema>>, predefined XML Schema documents

Each time a UML-schema is transferred, a package tagged with the <<Application Schema>> *stereotype* is mapped into an XML schema file (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. The schema components described in these files define the GML application schema of AFIS-ALKIS-ATKIS.

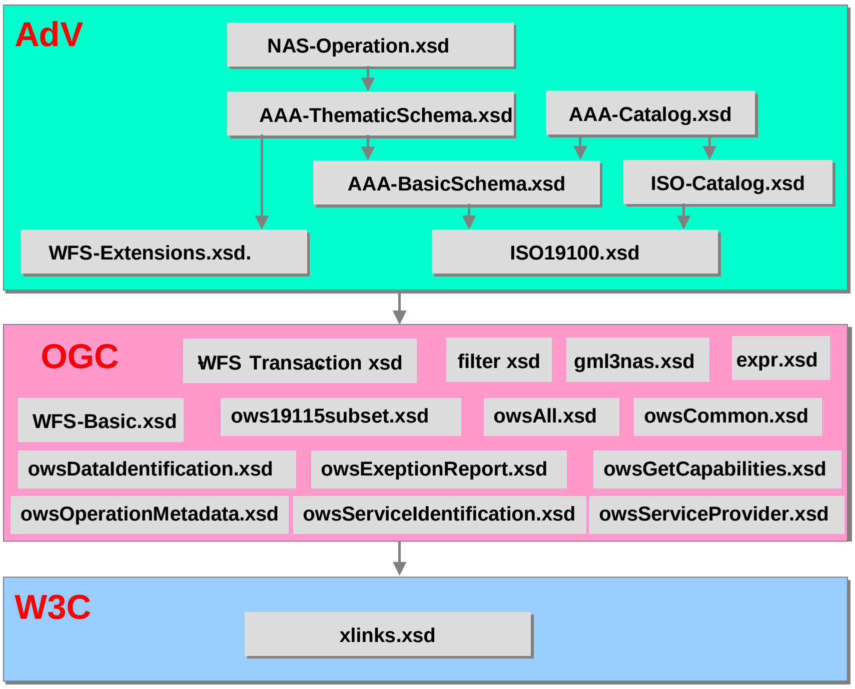


Figure 40: Basic structure of the central XML schemas of NAS

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/namespaces/adv/gid5.1" . 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-ThematicSchema.xsd" is to be replaced by a corresponding alternative GML application schema. 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, the description of the requests and responses of each operation are defined in the XSD file “NAS-Operations.xsd” for the individual operations to be run with NAS (see Item 10.2).

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*.

The extensions for WFS XML schemas and FES XML schemas described in item 10.2 are defined in a XSD file “WFS-Extensions.xsd”.

For exchanging of feature catalogues using XML (not part of the NAS) the XSD files “ISO-Catalog.xsd” and “AAA-Catalog.xsd” are derived from the AC-classes of the AAA basic schema and the FC-classes of ISO 19110.

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 (extract)

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

The mapping of the elementes shown in this table from UML type to the corresponding XML schema basic type is managed by the control parameter “ImplementedByType”. This constuction enables other thematic information schemas to define their own basic data types, that should be implemented as types with “simple content”.

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.

For supporting the translation of codes into corresponding descriptions the codelists of an application schema is implemented as a GML 3.0-dictionary.

External code lists

External code lists (<<ExternalCodeList>>) of an application schema 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

Abstract classes usually are mapped as *complexType*. Mixin classes are typically exceptions where their properties instead are mapped in instance-capable sub classes, because XML-schemas do not support the displaying of multiple inheritance. Doing this the "*copy down*" mechanism described in ISO 19118 is applied.

Abstract classes are applied in the schema tagged with attribute 'abstract="true"', see ISO 19118 A.5.2.10). The abstract classes not to be accepted into NAS are to be indicated for automatic NAS derivation though the relevant control parameters (“AbstractTypeUsage” = “false”).

In derived classes, the properties of the abstract top classes representatd in the XML schema are not accepted directly 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:element name="AA_Objekt" type="adv:AA_ObjektType"
          substitutionGroup="gml:_Feature" abstract="true"/>

    <xs:complexType name="AA_ObjektType" abstract="true">
          <xs:complexContent>
                <xs:extension base="gml:AbstractFeatureType">
                      <xs:sequence>
                            <xs:element name="lebenszeitintervall">
                                  <xs:complexType>
                                        <xs:sequence>
                                              <xs:element ref="adv:AA_Lebenszeitintervall" />
                                        </xs:sequence>
                                  </xs:complexType>
                            </xs:element>
                            <xs:element name="modellart">
                                  <xs:complexType>
                                        <xs:sequence>
                                              <xs:element ref="adv:AA_Modellart"
                                                          maxOccurs="unbounded"/>
                                        </xs:sequence>
                                  </xs:complexType>
                            </xs:element>
                            <xs:element name="anlass" type="xs:string" minOccurs="0"
                                  maxOccurs="unbounded">
                                  <xs:annotation>
                                        <xs:appinfo>
                                              <adv:referenzierteCodeList>
                                                  adv:AA_Anlassart
                                              </adv:referenzierteCodeList>
                                        </xs:appinfo>
                                  </xs:annotation>
                            </xs:element>
                            <xs:element name="zeigtAufExternes" minOccurs="0">
                                  <xs:complexType>
                                        <xs:sequence>
                                              <xs:element ref="adv:AA_Fachdatenverbindung"
                                                   maxOccurs="unbounded"/>
                                        </xs:sequence>
                                  </xs:complexType>
                            </xs:element>
                            <xs:element name="istTeilVon" type="gml:ReferenceType"
                                  minOccurs="0" maxOccurs="unbounded">
                                  <xs:annotation>
                                        <xs:appinfo>
                                              <adv:referenziertesElement>
                                              adv:AA_ZUSO
                                              </adv:referenziertesElement>
                                        </xs:appinfo>
                                  </xs:annotation>
                            </xs:element>
                            <xs:element name="inversZu_dientZurDarstellungVon"
                                   type="adv:InverseReferenceType"
                                  minOccurs="0" maxOccurs="unbounded">
                                  <xs:annotation>
                                        <xs:appinfo>
                                              <adv:referenziertesElement>
                                              adv:AP_Darstellung
                                              </adv:referenziertesElement>
                                              <adv:referenziertesElement>
                                              adv:AP_LTO
                                              </adv:referenziertesElement>
                                              <adv:referenziertesElement>
                                              adv:AP_FPO
                                              </adv:referenziertesElement>
                                              <adv:referenziertesElement>
                                              adv:AP_PTO
                                              </adv:referenziertesElement>
                                              <adv:referenziertesElement>
                                              adv:AP_LPO
                                              </adv:referenziertesElement>
                                              <adv:referenziertesElement>
                                              adv:AP_PPO
                                              </adv:referenziertesElement>
                                              <adv:inverseEigenschaft>
                                              adv:dientZurDarstellungVon
                                              </adv:inverseEigenschaft>
                                        </xs:appinfo>
                                  </xs:annotation>
                            </xs:element>
                      </xs:sequence>
                </xs:extension>
          </xs:complexContent>
    </xs:complexType>

    <xs:element name="AA_ZUSO" type="adv:AA_ZUSOType"
          substitutionGroup="adv:AA_Objekt" abstract="true"/>

    <xs:complexType name="AA_ZUSOType" abstract="true">
          <xs:complexContent>
                <xs:extension base="adv:AA_ObjektType">
                      <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 information about the cardinality of the attribut of the UML class is directly transformed in “minOccurs”- and “maxOccurs”-attributes of the element. That means that optional attributs of the conceptuel model (cardinality “0..n” with n=1,2,3,...*) are transformed as elements with “ninOccurs=0”. If an object instance do not carry this attribut, the corresponding element of the NAS is missing.

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 one or more appinfo-annotation element “adv:referencedElement” for evalution through NAS parsers. More tham one element always occur if the association links to an abstrct class that is not be represented in the XML-schema. In this case the instance-capable classes are dircetly derived.

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.

To simplify the request it is allowed in the FilterEncoding expressions to craete counter-relations in queries. For this purpose – and just for this purpose – the XML schema contains counter-relations by using the type “adv:InverseReferenceType”. The XML encoded object instances never contain counter-relations. For mor information see item 10.2.4 “Requirements for outputs”.

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>
                              <adv:referenziertesElement>
                                    adv:AX_Buchungsstelle
                              </adv:referenziertesElement>
                        </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, compositions 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 (in the 'substitutionGroup' of the element that carries the type of which the type of the feature is derived). 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="adv:TA_MultiSurfaceComponent"/>
```

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 10.2). 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.

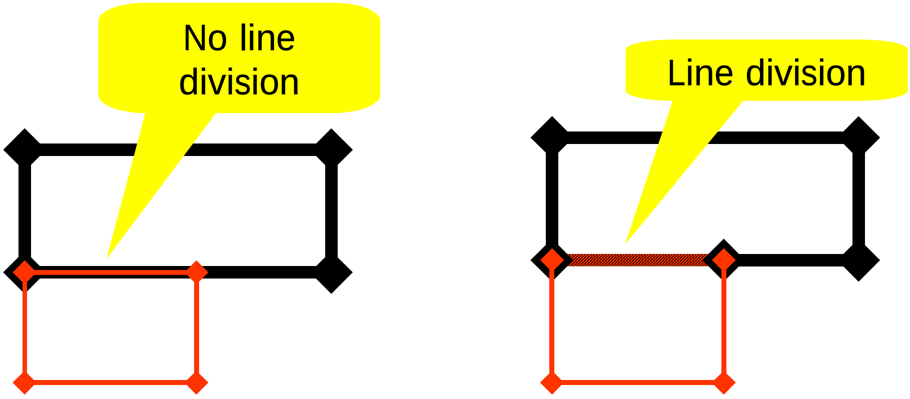


Figure 41: Explanation of line division

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.

For clarification both situations of figure 10-5 are encoded. Firs the left situation (no line division):

```
...
<AX_Flurstueck gml:id=“DEBY000000000001”>
  <!-- ... -->
  <position>
    <gml:Surface>
      <gml:patches>
        <gml:PolygonPatch>
          <gml:exterior>
            <gml:Ring>
              <gml:curveMember>
                <gml:Curve>
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:pos>3601085.954 5943996.138</gml:pos>
                      <gml:pos>3601085.954 5943998.138</gml:pos>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve>
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:pos>3601085.954 5943998.138</gml:pos>
                      <gml:pos>3601078.954 5943998.138</gml:pos>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve>
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:pos>3601078.954 5943998.138</gml:pos>
                      <gml:pos>3601078.954 5943996.138</gml:pos>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve>
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:pos>3601078.954 5943996.138</gml:pos>
                      <gml:pos>3601085.954 5943996.138</gml:pos>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
            </gml:Ring>
          </gml:exterior>
        </gml:PolygonPatch>
      </gml:patches>
    </gml:Surface>
  </position>
  <!-- ... -->
</AX_Flurstueck>
...
<AX_Gebaeude gml:id=“DEBY000000000002“>
  <!-- ... -->
  <position>
    <gml:Surface>
      <gml:patches>
        <gml:PolygonPatch>
          <gml:exterior>
            <gml:Ring>
              <gml:curveMember>
                <gml:Curve>
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:pos>3601082.954 5943994.138</gml:pos>
                      <gml:pos>3601082.954 5943996.138</gml:pos>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve>
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:pos>3601082.954 5943996.138</gml:pos>
                      <gml:pos>3601078.954 5943996.138</gml:pos>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve>
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:pos>3601078.954 5943996.138</gml:pos>
                      <gml:pos>3601078.954 5943994.138</gml:pos>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve>
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:pos>3601078.954 5943994.138</gml:pos>
                      <gml:pos>3601082.954 5943994.138</gml:pos>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
            </gml:Ring>
          </gml:exterior>
        </gml:PolygonPatch>
      </gml:patches>
    </gml:Surface>
  </position>
  <!-- ... -->
</AX_Gebaeude>
...
```

For comparision now the right situation, where the lower edge of the parcel is divided into two edges in order to get identical geometry:

```
...
<verwendeteInstanzenthemen>Flurstücke und
Gebäude</verwendeteInstanzenthemen>
...
<AX_Flurstueck gml:id=“DEBY000000000001“>
  <!-- ... -->
  <position>
    <gml:Surface>
      <gml:patches>
        <gml:PolygonPatch>
          <gml:exterior>
            <gml:Ring>
              <gml:curveMember>
                <gml:Curve>
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:pos>3601085.954 5943996.138</gml:pos>
                      <gml:pos>3601085.954 5943998.138</gml:pos>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve>
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:pos>3601085.954 5943998.138</gml:pos>
                      <gml:pos>3601078.954 5943998.138</gml:pos>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve>
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:pos>3601078.954 5943998.138</gml:pos>
                      <gml:pos>3601078.954 5943996.138</gml:pos>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve>
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:pos>3601078.954 5943996.138</gml:pos>
                      <gml:pos>3601082.954 5943996.138</gml:pos>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve>
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:pos>3601082.954 5943996.138</gml:pos>
                      <gml:pos>3601085.954 5943996.138</gml:pos>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
            </gml:Ring>
          </gml:exterior>
        </gml:PolygonPatch>
      </gml:patches>
    </gml:Surface>
  </position>
  <!-- ... -->
</AX_Flurstueck>
...
<AX_Gebaeude gml:id=“DEBY000000000002“>
  <!-- ... -->
  <position>
    <gml:Surface>
      <gml:patches>
        <gml:PolygonPatch>
          <gml:exterior>
            <gml:Ring>
              <gml:curveMember>
                <gml:Curve>
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:pos>3601082.954 5943994.138</gml:pos>
                      <gml:pos>3601082.954 5943996.138</gml:pos>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve>
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:pos>3601082.954 5943996.138</gml:pos>
                      <gml:pos>3601078.954 5943996.138</gml:pos>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve>
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:pos>3601078.954 5943996.138</gml:pos>
                      <gml:pos>3601078.954 5943994.138</gml:pos>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve>
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:pos>3601078.954 5943994.138</gml:pos>
                      <gml:pos>3601082.954 5943994.138</gml:pos>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
            </gml:Ring>
          </gml:exterior>
        </gml:PolygonPatch>
      </gml:patches>
    </gml:Surface>
  </position>
  <!-- ... -->
</AX_Gebaeude>
...
```

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/5.1"
  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/gid/5.1 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 accomplished 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 chapter 11 must be used.

For NAS documents that contain objects in a „FeatureCollection“ the standard reference system has to be applied in the attribute “srsName” of “gml:envelope”.

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>
```

Encoding of units of measures in the NAS

Principally, each figure of the NAS file representing an unit of measure (e.g. length, area, angle) have to refer to an unit of measure. This reference is done by applying an URI (Uniform Resource Identifier) at the attribut “uom”. The syntax and the defined association listed in chapter 11 have to be used.

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 to be exported in accordance with the standard regulations is to not be defined in NAS. Examples: AP_GPO, AX_Request (de : AX_Auftrag).
- 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 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_Object), the direct representation is nearby. In this case, attribute "position" should be linked as a local property element with a *choice* element of the permitted "*geometry types*" ("gml:point", "gml:curve", "gml:multiCurve" and "gml:surfac").

- 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. This is the default case (TRUE).
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.
ExtendsType	This parameter is used for complexTypes which shall extend a predefined XML-schema-Type by the xsd:extension-mechanism.
ImplementedBy	Indicates which XML schema type implements an UML class.

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.

Table: Overwiev about the defined control parameters for packages with stereotype <<Application Schema>>

Name	Meaning
targetNamespace	XML namespace, e.g. "http://www.adv-online.de/namespaces/adv/gid/5.1"
xmlns	XML namespace abbreviation, e.g. "adv"
File name	Name of the XSD file, if it is different from the name of the package
ISO19100	"True", if the contents of the packages are part of ISO 19100

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. By representing of GM_Curve in AFIS, ALKIS and ATKIS data, gml:Curve with just one gml:LineStringSegment has to be used. if the GM_Curve has linear interpolation in all segments. (gml.LineString must not be used in these cases and its application is just allowed in Filter Encoding expressions).

3. By representing GM_PolyhedralSurface in AFIS, ALKIS and ATKIS data gml:surface with just one gml:PolygonPatch has to be used. (gml.Polygon must not be used in these cases and its application is just allowed in Filter Encoding expressions).

4. Because the majority of area-objects (e.g. parcels), which are defined as GM_MultiSurface consist only of one area, 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 in AFIS. ALKIS and ATKIs data gml:Ring with exact one gml:Curve must be used if the GM_Ring consists of only one GM_Curve. (gml.LinearRing must not be used in these cases and its application is just allowed in Filter Encoding expressions).

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 (and raplaced in GML 3.2 by an other machanism).

8. The standard properties of GML features “gml:name” and “gml:description” are only allowed in GML dictionaries, but not for properties of features in namespace of the NAS.

The annex of this document contains a complete representation of the basis schema in GML.

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.

For usage in thematic information systems three general operations are specified for updating data of the primary data base, for request of outputs of these data and for general information about the properties of primary data base stoage.

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)
- Descovery if general information about the properties of a primary data base storage.

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 are automatically 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. The AAA-Basic Schema.xsd (de: AAA-Basisschema.xsd) contains the common usable basic operations

The opreations are based on the OWS Common Implementation Specification 1.0 that must be used. Specifically, each NAS implemenation must support the GetCapabilityOperation.

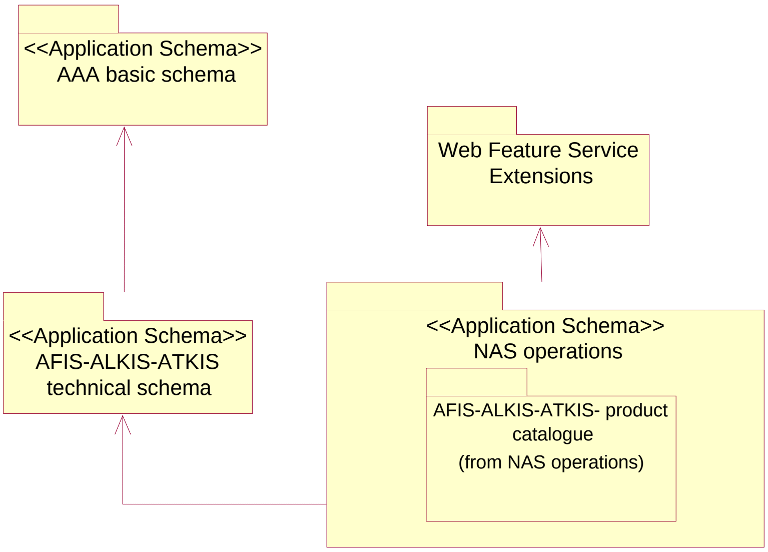


Figure 42: UML-package "NAS-operations” in the context of application schema components

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 Open Geospatial 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[2]. 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[3] :

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="#DEXXXX00000002"/>
</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>
```

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

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

- 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>`[4] . The WFS-Extensions.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).

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

- 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-Extensions.xsd is supplemented by the following definitions:

```
<xs:complexType name="XlinkPropertyPathType" mixed="true">
   <xs:complexContent mixed="true">
      <xs:extension base="ogc:PropertyNameType">
         <xs:attribute name="leafOnly" type="xs:boolean" default="false">
            <xs:annotation>
               <xs:documentation>Das Attribute regelt, ob alle Objekte
      längs des Pfades oder nur das Ziel des Pfades selektiert werden. Default ist
      das bis zur GeoInfoDok 4.0 benutzte Standardverhalten (alle Objekte entlang
      des Pfades).</xs:documentation>
            </xs:annotation>
         </xs:attribute>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="XlinkPropertyPath" type="adv:XlinkPropertyPathType"
   substitutionGroup="ogc:PropertyName">
   <xs:annotation>
      <xs:documentation>
      Dieses Element darf in der NAS ein ogc:PropertyName Element ersetzen,
      allerdings nie in einem ogc:Fliter Element, nur als direktes Kindelement
      eines wfs:Query Elements.
      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“.
      Als Wert wird eine Xpath-Pfadangabe verwendet, bei der am Ende ein
      Objekt steht, bei dem die Auflösung abbricht.
      Beispiel: Ein XlinkPropertyPath
        "istGebucht/AX_Buchungsstelle/istBestandteilVon/AX_Buchungsblatt"
      bei einer Query auf AX_Flurstueck führt dazu, dass die Buchungsstelle und das
      Buchungsblatt in der Ergebnismenge für jedes selektierte Flurstück direkt
      mit zurückgeliefert werden.
      </xs:documentation>
   </xs:annotation>
</xs:element>
```

- Application of `<ogc:PropertyIsOfType>` for descovery ot a object property. For properties with comlexContent this is the qualified element name of the child element, for properties with simpleContent it is the qualified type name of the element name.

  Therefore the WFS-extensions is extended by following definitions:

```
   <xsd:element name="PropertyIsOfType" type="ogc:BinaryComparisonOpType"
      substitutionGroup="ogc:comparisonOps">
      <xsd:annotation>
         <xsd:documentation>Der Vergleichsoperator ermittelt den Typ der im ersten
Operanden angegebenen Eigenschaft (ogc:PropertyName) und vergleicht ihn mit der Angabe
im zweiten Operanden (ogc:Literal). Beide Angaben müssen identisch
sein.</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), <replace> (amend, overwrite object) and <delete> (delete object). In the feature catalogues is to specify which changes result in <replace> 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:Transaction>
  <adv:Replace>
    <adv:AX_Flurstueck">
     ...
     <!-- Inhalt wie bei Insert -->
     ...
    </adv:AX_Flurstueck>
   <ogc:Filter>
    <ogc:FeatureId fid="DEBY0000F000000220010101T000000Z"/>
   </ogc:Filter>
 </adv:Replace>
</wfs:Transaction>

  <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 <replace> operations may contain only *FeatureId* elements. More complex filter criteria are not permitted.
- In *FeatureId* conditions in "<replace>" 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. If there are more than one versions of one feauture the object identifier is also extended by creation date and creation time withot separation sign, in order to avoid multiple ID in one request. For this rule there is the following exception: Within an update request with more than one <replace>-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.
- <replace>-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 set-up and updating of metadata is not yet part of the modelling or definitions. This will follow after determination of the thematic requirements.

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.

<replace>:
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 <*replace*> 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.

<replace>
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. The NAS only mentions the role of the association that is determined in the aplication schema as navigable direction. Counter relations are not allowed in the NAS. Nevertheless it is allowed to select features that are connected via counter relation with an other feature in order to simplify Filter Encoding expressions. In this case the explicitly named role name of the conter relation must be used if it exists. If it does not exist the role name of the navigable direction is used and extended by „inversTo“.

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*>, <*replace*> 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.

10.2.10 Descovery of properties of primary data stirage

Each softare component that realizes a NAS data exchange interface, must support the GetCapabilities-operation.

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 -<replace> 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 <*replace*> (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 <*replace*> 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 the database creates the geometrical and topological consistency by itself ("**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	<*replace*> 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	<*replace*> 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)

10.5.2.3 Portioning of NBA- data

The portioning of NBA- data is optional. There is no duty to use it, but it allowes the user of NBA- data to get the data step by step. This issue formerly had a good acceptance in former applications (e.g. ALK) by transferring large data files.

Especially the following requirements are addressed: Providing of the ALKIS and ATKIS data in geometrical portions. The size of a portion shall be variable but unique for one user (parameter of the portion size is indicated in user profiles). Non spatial features (NREO) and combined features (ZUSO) are selected for each portion by associations mentioned in selection criteria in the user profile. Portioning considering just the data amount is not sufficient.

10.5.2.3.1 Formal form of the portioning

The processing of portioning is done with the AX_UserRelatedDataUpdating_NBA_Documents that are produced by the system in an automatical NBA-process. These are created at a specific time considering the requirements of AX_UserGroupNBA. The number of used NBA-documents for processing the request is left to the implementation, but there must be a summerised request protocol for the one ore more result files. Because of [1..n] utilization requests with code

“AX_UserRelatedDataUpdating_NBA” (0040), all necessary portions are transfered as independant updating requests[5].

For redundant free transfer of features, explanations follow at the parametrising of the portioning (see below).

All portions of an utilization request contain the same application number and the same request number. For following deliveries the request number arises.

For each portion, metadata must be created that show at least the geometrical and logical area of the portion (that will be added after determination of the technical requirements in the field of metadata).

10.5.2.3.2 Reqirements for the sending system

10.5.2.3.2.1 Parametrising of Portioning

On the level of user profile of the NBA process the portioning can be activated by using the portioning parameter or deactivated by ignoring. The selection an prtioning is done in i two step process:

1. The selection criteria in AX_UserGroupNBA indicate the selection of the objects to be totaly transfered by the delivery (independantly, if there is a portioning or not).

2. The separation of the selected objects in portions is done by portion parameters as described below. It is necessary to define reasonable portioning parameters by creation of the NBA-profile.

If there are no update data in a portion of the corresponding delivery, it is not necessary to create this portion. This way, the amount of portion per delivery is varable, but it becomes clear by determination of the file name, how comprehensive the total delivery is.

The following portion parameter can be used: Indication of the side length in meter. It it is applied, it is a positive integer figure differnt from null. The delivering system automatically separates the area into corresponding squares mentioned by the selection criteria. Hereby, the following rules must be regarded:

[5] In this case the term „updating request“ has not the same meaning as in AX_UpdatingRequest, but means the response AX_UserDefinedDataUpdating_NBA that contains the same WFS operation „transaction“ as in the updating request.

- The area is firstly treated from west to east, then from south to north. The first left buttom corner can be calculated from the most south-westerly point to the next coordinate pair with complete meters, that is south-west of it. If the most south-westerly point if the delivery area contains already a pair of coordinates in complete meters, this point will be directly be used.

- All spatial objects (REO) that are located in a portion square and all additional non spatial objects (NREO) and combined objects (ZUSO) that have been requested by selection criteria and associated to the corresponding spatial objects (REO) must be in one portion.

- If one object would apply in more than one portions due to its extension, it is just delivered with the first one of the series of squares. Example: A surface object extends upon portion (=square) 12, 13, 21 and 22. It is delivered in portion 12.

- Attached NREO and ZUSO are just delivered in that portion in which they appear at first.

10.5.2.3.2.2 Bracketing of the delivered portions

All portions of a delivery are indicated as belonging together by sufficient portion codes. These code must be applied

- in the attribute “AX_UserRelatedDataUpdating_NBA.portionCode

- in the file name.

The portion code is created as follows:

<NBA-ProfileCode[6]>
<_>
<date of NBA creation in format DateTime[7]>
<_>
<continous number of portion, no leading nulls>
<of>
<total amount of portions of delivery, no leading nulls[8]>
<_>

[6] The profile code should be a clear and short term. For file names with problematic characters (e.g. space) are avioded as they are not accepted by derivation of this name parts out of the profile code.

[7] Context: Daily actuality of the data base. For more often updates, an extension by a corresponding exact time is necessary, that is applied at AX_User in attribute “lastDelivery” (exaxt system time of the delivery)

[8] Total amount is calculated seperately for geometric and non-geometric portions.

<pair of coordinates of the lower left corner of the corresponding portion, separated by „_”[9]>

The porpose of this syntax is a) the bracking of portions of one delivery and b) to enable the user to map the portions spatially without looking into the NBA-document.

Finaly file name are created as follows:

- *Firmaxy_041201_124von211_3401559_5572720.xml*
- *RMR_040715_7von31_3401449_5573000.xml*

Example:

The selection criteria of AX_UserGroupNBA indicate:

```
<wfs:Query typeName="AX_PunktortTA">
    <adv: XlinkPropertyPath>istTeilVon/AX_Grenzpunkt</adv:XlinkPropertyPath>
    <ogc:Filter>
    <ogc:BBOX>
    <ogc:PropertyName>position</ogc:PropertyName>
    <gml:Envelope srsName="urn:adv:crs:DE_DHDN_3GK3">
    <gml:pos>3353100.000 5532300.000</gml:pos>
    <gml:pos>3353300.000 5532500.000</gml:pos>
    </gml:Envelope>
    </ogc:BBOX>
    </ogc:Filter>
</wfs:Query>
```

1. The additional portion parameters indicate:
   length = 100m

2. Result:
   In maximum 4 portions will be created, containing AX_PointLocationTA an corresponding AX_BoundaryPoint

In order to enhance the overwiev, reasonable file indicators can be generated by profile code and date.

10.5.2.3.3 Reqirements for the addresed system - Processing of the delivery

Using the profile code ("3of8") it is checked if all potions of ane delivery have been implemented. The addressed system must know the portions of the total delivery. Only after a complete implementation of a NBA-delivery it can be receipted by request and it is then allowed to sart the next delivery.

10.5.2.4 Receipt of NBA-deliveries

After implemenation of a NBA-delivery, a receipt can be sended to the delivering institution by NBA-receipt-request “AX_NBAReceipt”, if this is state-specificly required

[9] Full meter accuracy, that means no komma; example"3401559_5572720"

and requested by the NBA-delivery. The kind of processing of the NBA-receipt has to be defined state-specific.

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:

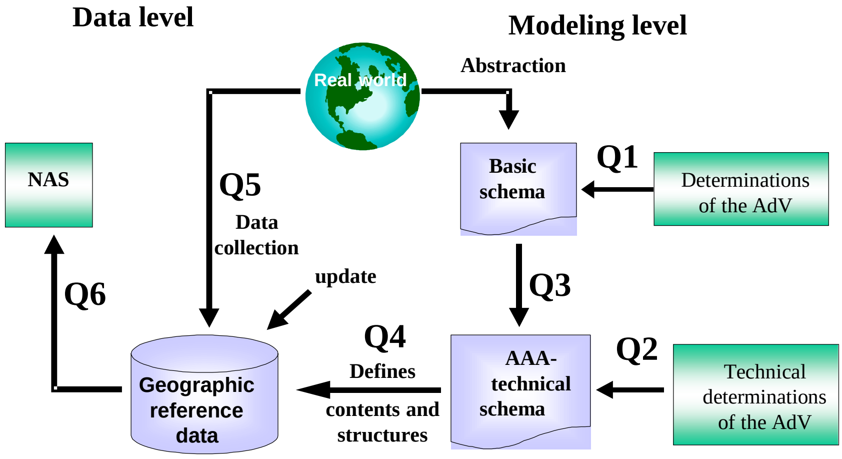

Data level
Modeling level
Real world
Abstraction
NAS
Q5
Data collection
Basic schema
Q1
Determinations of the AdV
Q6
update
Q3
Q4
Defines contents and structures
Geographic reference data
AAA-technical schema
Q2
Technical determinations of the AdV

Figure 43: 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.	**geoinformation**
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 Geospatial 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