AFIS
ALKIS
ATKIS

Documentation on the Modelling of Geoinformation of Official Surveying and Mapping

(GeoInfoDoc)

Main Document

Version 7.0.2
Date: 10.05.2015

Working Committee of the Surveying Authorities of the States of the Federal Republic of Germany

Table of contents

Document History

Version	Updated	Comment	Parties involved
7.0.0	20.07.2014	Adaptations in line with revisions list (e.g. instance themes) and editorial revision	M.Seifert
7.0beta	02.07.2014	Amendments based on comments by Joachim Stiebler (BW)	J. Stiebler, M.Seifert
7.0beta	07.04.2014	Editorial revision	M. Seifert
7.0beta	20.03.2014	Amendments based on comments by Stefan Schliebner (RP)	AAA-Maintenance Group
7.0beta	13.03.2014	Complete revision based on revision notifications, changeover to Enterprise Architect (EA) and amended modelling guidelines	PG GDI-Standards

Normative References

The following referenced documents are indispensable for the implementation of the AAA application schema. For sources with a specified date, only that one named version applies. For sources where no date is specified, the most recently published version including any revisions always applies.

ISO/IEC 19501:2005, Unified Modeling Language Specification (UML), http://www.uml.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, 2 May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

XML Schema Part 2: Datatypes-W3C Recommendation, 2 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/

Web Feature Service (WFS) 1.0, Open Geospatial Consortium, 2002 http://www.opengis.org/techno/specs/02-058.rtf

OWS Common Implementation Specification 1.0 Open Geospatial Consortium, 2005 https://portal.opengeospatial.org/files/?artifact_id=8798

OpenGIS Implementation Specification Filter Encoding 1.0.0 (http://www.opengis.org/techno/specs/02-059.pdf)

OpenGIS Implementation Specification Simple Features for SQL 1.1 (http://www.opengis.org/techno/specs/99-049.pdf)

Xpath (http://www.w3.org/TR/xpath)

ISO 19103 Geographic information: Conceptual schema language, 2005, http://www.isotc211.org/Outreach/Overview/Overview.htm

ISO 19107 Geographic information: Spatial schema, 2003, http://www.isotc211.org/Outreach/Overview/Overview.htm

ISO 19108 Geographic information: Temporal schema, 2002, http://www.isotc211.org/Outreach/Overview/Overview.htm

ISO 19109 Geographic information: Rules for application schema, 2005, http://www.isotc211.org/Outreach/Overview/Overview.htm

ISO 19110 Geographic information: Methodology for feature cataloguing, http://www.isotc211.org/Outreach/Overview/Overview.htm

ISO 19111 Geographic information: Spatial referencing by coordinates, 2009, http://www.isotc211.org/Outreach/Overview/Overview.htm

ISO 19115 Geographic information: Metadata, http://www.isotc211.org/Outreach/Overview/Overview.htm

ISO 19118 Geographic information: Encoding, 2005, http://www.isotc211.org/Outreach/Overview/Overview.htm

ISO 19123 Geographic information: Schema for coverage geometry and functions, 2005, http://www.isotc211.org/Outreach/Overview/Overview.htm

- ISO 19127 Geographic information: Geodetic codes and parameter, 2005, http://www.isotc211.org/Outreach/Overview/Overview.htm
- ISO 19135 Geographic information: Procedures for item registration, 2005, http://www.isotc211.org/Outreach/Overview/Overview.htm

ISO 19136 Geographic information: Geography Markup Language, 2007, http://www.isotc211.org/Outreach/Overview/Overview.htm

ISO/TS 19139 Geographic information: Metadata - XML schema implementa-tion, 2007, http://www.isotc211.org/Outreach/Overview/Overview.htmISO 19157 Geographic information: Data Quality, http://www.isotc211.org/Outreach/Overview/Overview.htm

1 Structure, Content and Objective

1.1 Initial Situation, Motive and Objectives

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

The concepts according to which ALB, ALK and ATKIS were founded originated in the 1970s and 1980s. They still serve, today, as the platform on which the relevant geospatial reference data inventories are created and maintained. 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 current ALK and ALB Information systems will therefore be operated as integral parts of **ALKIS**(Authoritative Real Estate Cadastre Information System), enabling the further development of 3D geospatial reference data to be addressed. There is the need, particularly in connection with buildings in ALKIS, to be able to store 3D information. 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.

There is an AdV standard available for **Digital Ortho photos** (DOP), that, according to current understanding, is not an application that is part of the application schema, but will still be incorporated into the overall documentation under the heading *Photo Based Data* in chapter 2 of *The AFIS-ALKIS-ATKIS reference model and Product Groups* .

The derivation of **3D town and landscape models** from the geospatial reference data will be enabled by a combination of 3D Information in ALKIS and the DGM in ATKIS plus terrain texturing.

In order to use the geographic information system characteristics defined for the 2D range, the basic classes for the modelling of 3D data will be integrated in the basic schema, In this way the ALKIS update process can also be used for the cost effective updating of 3D geospatial reference data.

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

The AdV projects AFIS, ALKIS and ATKIS, with their nationally standardised features, are jointly 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

Core data (Grunddatenbestand) is 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-related updating of secondary databases** (NBA). States, which 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. Regarding **history management**, the effective-date-referenced storage of the respective data inventories is considered sufficient,

The integration of 3D Information in the general application schema for AFIS, ALKIS and ATKIS means that the need for an versioning and historicization concept is satisfied.

1.3 Target Groups and Users

Regarding the content and structuring of geospatial reference data and the efficiency of its specification, countrywide users and GIS industries demand a unified, nationwide core data inventory. In a holistic view of the official Surveying, the core data inventory of AFIS, ALKIS and ATKIS should be combined to form this core data inventory of the geospatial data of official Surveying

Within the European framework guidelines for construction of a geospatial infrastructure in Europe, **INSPIRE**, the standards conformant modelling of geospatial reference data plays a significant role. A central goal of INSPIRE is the provision of more and, above all, consistent geospatial data for community policies and their implementation in member nations at all levels. The emphasis here is on environmental policy. In a European geospatial data infrastructure the geodata itself can have different degrees of harmonisation within a technical management area. For this reason INSPIRE contains three distinct appendices that relate to three different topic areas of geospatial data that are required for a wide spectrum of environmental policy measures. Dependent on whether the geospatial data is to be used for geospatial referencing of other data or whether harmonised geospatial data is required for political measures with a direct or indirect effect on the environment, and dependent on the degree of harmonisation that has already been achieved within the community, there are different objective-deadlines for meeting the requirements of INSPIRE together with various harmonisation procedures. How this geospatial data should be organised and harmonised is not regulated here but in the technical implementation stipulations

The goal of INSPIRE is not to provide a comprehensive programme for the collection of new geospatial data in the member states. Rather the documentation of existing geospatial data is desired, in order to optimise the use of data that is already available. For this purpose, services (Web Services) will be defined that make the geospatial data more easily available and more interoperable and there will be attempts made to solve the problems associated with the usage of geospatial data (access rights, price, etc.). In this way, INSPIRE will smoothe the way for a step by step harmonisation of geospatial data in the member states. With the AFIS-ALKIS-ATKIS application schema, the AdV is already amply prepared for INSPIRE conformant data release.

GIS and CAD users are very interested in the construction of 3D models that build on the official real estate cadastre data, in order to be able to display and better visualise their plans based on these official foundations. Furthermore, a uniform 3D model based on the GeoInfoDoc can be a suitable platform for storage of the incidental 3D information. Currently there is no official certification for this information.

The **EU Directive** for reducing environmental noise (2002/49/EG) stipulates the future, regular, detailed noise propagation calculation that can only be based on a continuously updated 3D models of towns. The 3D information built on GeoInfoDoc offers the foundation for the determination of environmental noise, offers update mechanisms and enable the required, regular checking of noise categories through the use of versioning / historicization concepts.

For the **migration** from the current situation, certification is a basic procedure in connection with a staged concept. The detailed planning behind of the migration design is to be handled by each Federal State individually. A reverse migration into the interfaces of the existing systems for a transitional provision of data to the user is probably required for the foreseeable transition period. The migration concept has only a temporary importance and therefore is 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 (optionally also enhancements of 3D information),
- 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).
- digital image models (DBM) in digital Ortho photo form (DOP).

The content, structures and production instructions for the products of the reference model are defined at **control level** through the feature catalogue (OK) and portrayal 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 portrayal and cartographic layout of the object,
- instructions for the arrangement 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 encompasses the formation of presentation and map geometry objects and therefore also includes the process of cartographic generalisation.

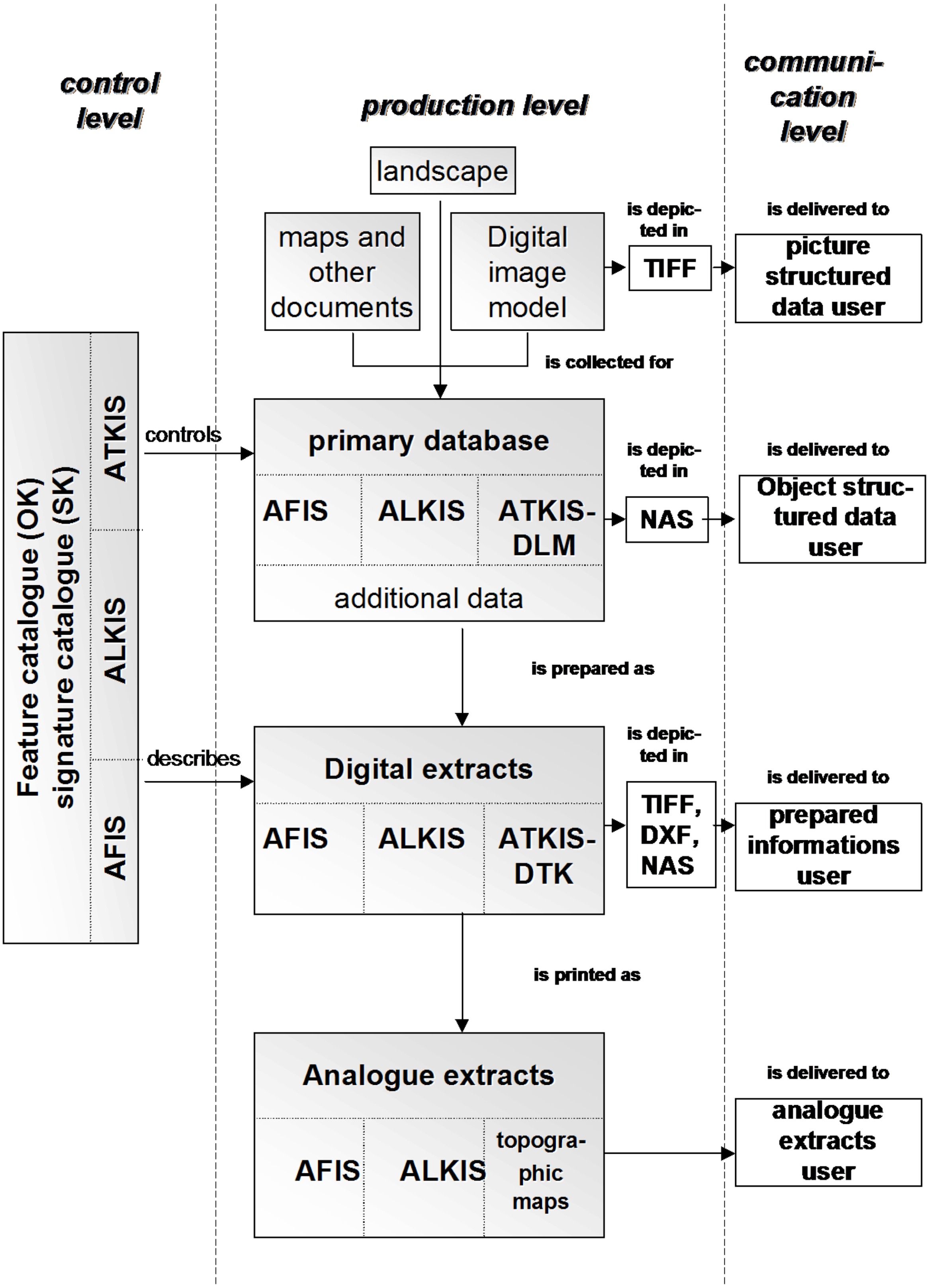


Figure 1: Common AFIS-ALKIS-ATKIS Reference Model (Source: The common AFIS-ALKIS-ATKIS Reference Model, 1996)

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 **features** with their semantic and geometric information and also the **additional data**required for presentation purposes:

- namely the **presentation objects** for text and signatures, and
- the map geometry objects with the associated **map geometry** for a certain map scale that are linked to the topographical objects through a unidirectional relation.

The data of the primary databases contains the complete description of features, including the data on their cartographic or textual portrayal 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 the **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 of the AAA Basic Schema

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 by 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 also uses parts of the OGC specifications. CityGML forms the foundation of the integration of 3D information (OGC Best Practices Document, version 0.4.0)

3.1.2 Modelling and Description Language

The AdV decided to use the *Unified Modelling 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. UML semantics and notation 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 the type of implementation and the programming language used. 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 laid down in the corresponding ISO standards 19118 *Encoding* and

19136 *Geography Mark-up Language (GML).* The language XML (Extensible Mark-up Language) of the World-Wide-Web-Consortium (W3C) is used as the format.

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. A fundamental concept used in abstracting the real world is the introduction of the feature and the rules about how it is recorded and updated. Features are classified by type. At the type level, the application schema describes the feature types of the real world. Data exists at the instance level. It represents individual instances 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 to document the data content for a specific application environment so as to obtain unique information about that data.

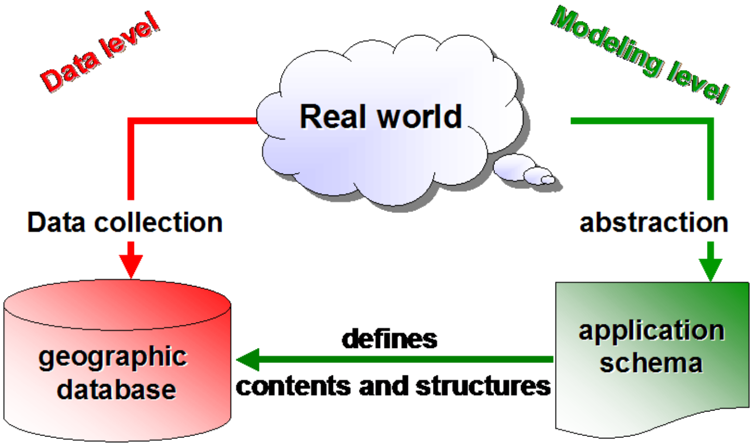


Figure 2: The role of the application schema

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

An application schema can use specifications from several sub-schemas. In the case of the AFIS-ALKIS-ATKIS application schema, mainly sub-schemas of the ISO 19100 series of standards are used. In those areas where, until now, there are no ISO standards, additional schemas of the Open Geospatial Consortium are used.

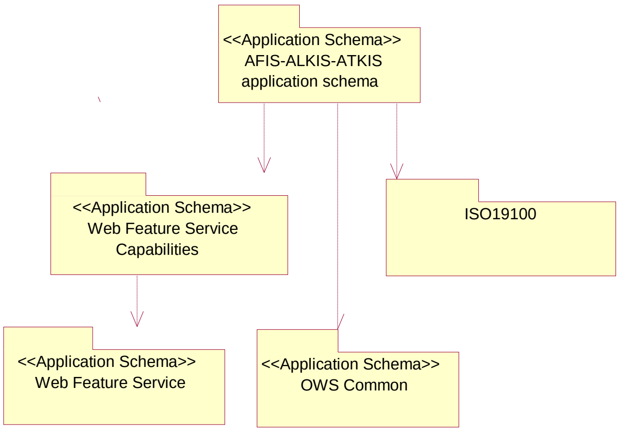


Figure 3: Dependency of the AFIS-ALKIS-ATKIS application schema on the structures standardised by ISO 19100 and OGC

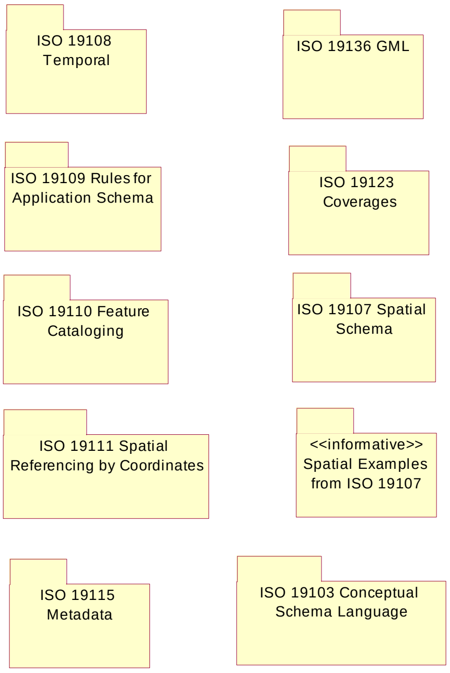


Figure 4: Components of the ISO 19100 series of standards used

The AFIS-ALKIS-ATKIS application schema is sub-divided into the

- basic schema
- ALKIS-ATKIS-AFIS thematic (domain) schema
- AAA versioning schema
- NAS operations

The basic schema is the basis on which features are modelled in the domain schemas. The versioning schema shows the concept for historicizing features. An internal schema is not a component of common modelling, it is the result of mapping the conceptual application schema onto specific GIS systems during the course of the implementation process.

The application schema is the basis on which operations for data exchange and the technical (domain) specifications for outputting data are ultimately defined.

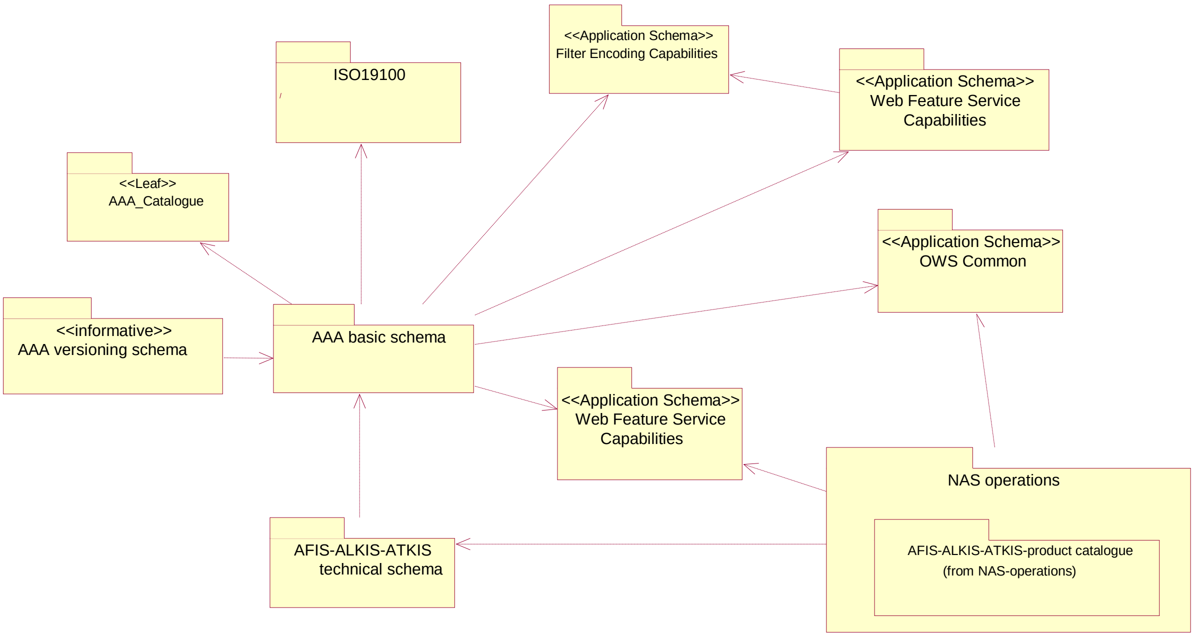


Figure 5: The components of the AFIS-ALKIS-ATKIS application schema

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.

The extension to the AAA basic schema classes is required for deploying a 3D domain schema whose basic schema, until now, did not contain geometry types for describing volumetric objects.

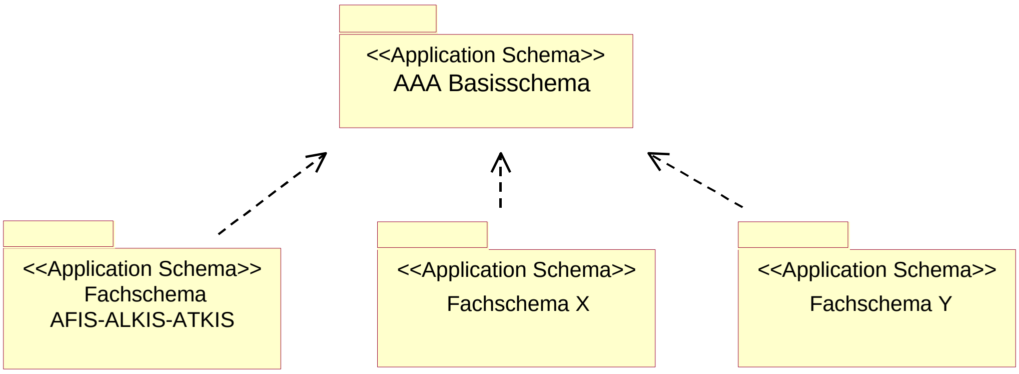


Figure 6: The basic schema as a basis forapplication-specific domain schemas (e.g. AFIS, ALKIS and ATKIS)

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

The packages AAA_Nutzerprofile (UserProfiles) and AAA_Operationen (Operations) only serve for anchoring user management or operations modelling, respectively, in the basic schema. They only comprise empty, abstract classes, which have be concretises in the respective thematic schemas. Forthat reason, a further description of these packages is omitted.

The following systematic 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 extensions to the standardised *Feature Catalogue* are prefixed with AC
3. Classes with fundamental meaning for AFIS, ALKIS and ATKIS are prefixed with AA
4. Classes derived from the ISO TS_*Component classes ("simple topology"), are prefixed with TA; also the analogously created class for topological

surfaces with multiple spatially separated geometries (TA_MultiSurfaceComponent)

5. Classes with commonly used geometries are prefixed with AG
6. Classes of independent geometries are prefixed with AU
7. Classes of presentation objects are prefixed with AP
8. Classes for the modelling of PointCoverages are prefixed with AD

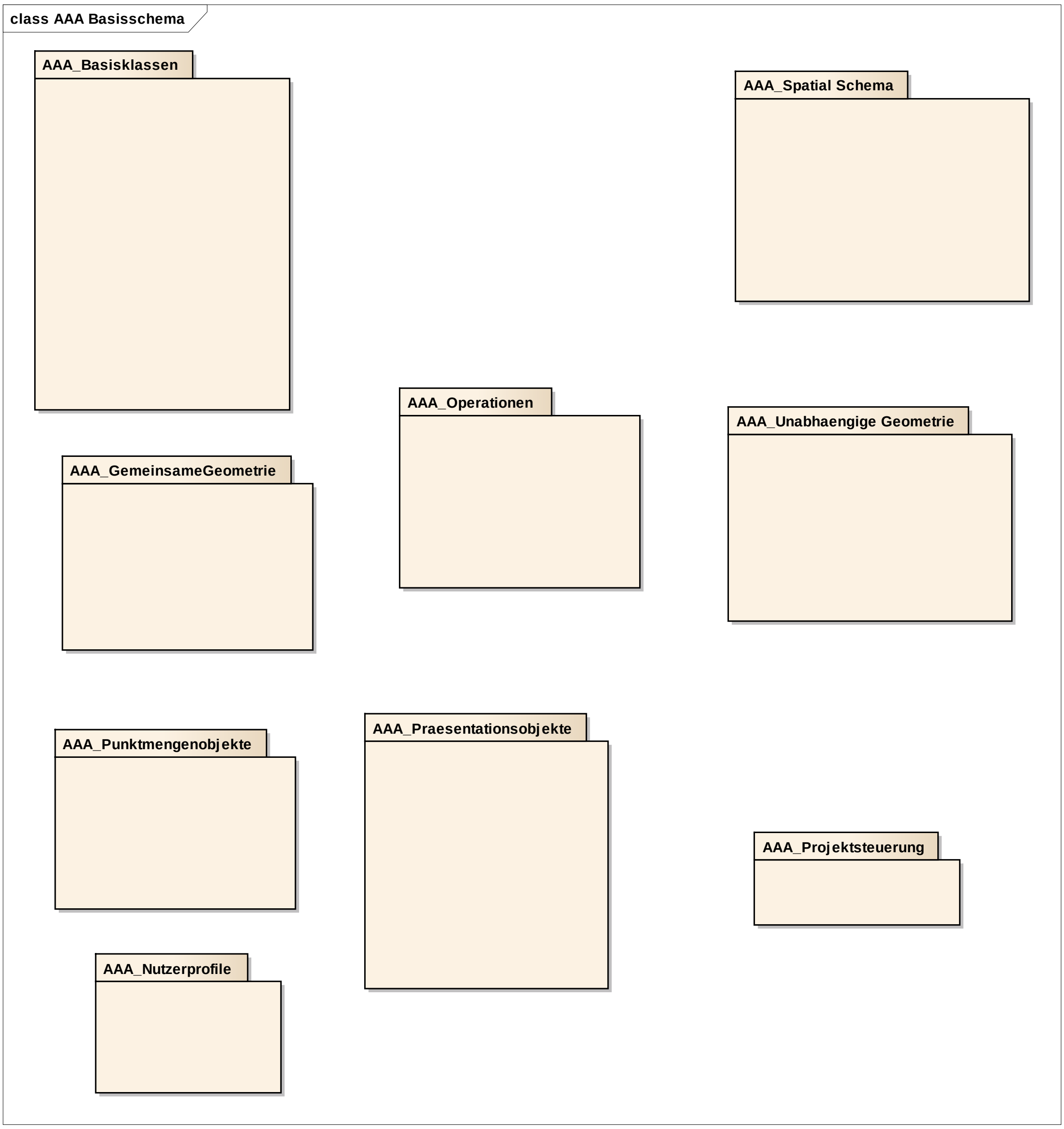


Figure 7: Components of the AAA basic schema

3.3.1 Fundamental Principles of Object Formation

The rules for creating application schemas are defined in the ISO 19109 standard *"Rules for Application Schema"*, developed by ISO/TC 211. This standard also contains the general model for describing and forming features *(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 be point, line or surface geometries or be of type PointCoverage. **Objects without spatial reference** (e.g. people) bear no geometry and cannot be fixed to 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, the following 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 should also be shown.

- Non-spatially-referenced elementary objects (AA_NREO)

  Non-spatially-referenced elementary objects are to be formed if only technical and 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 features that consist of a large number of geometric locations and each with the same attribute types for each location (e.g. Digital Terrain Models, temperature or pressure distribution), it is more efficient to use coverage objects instead of single REOs for each point. A Coverage Object maps a set of (function which maps) geometries to their respective attribute values.

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

The management of the **history of objects** is supported. **Integration and interconnection of features with the technical data** of other technical areas are also supported.

All instantiable thematic object classes are to be derived in the application-referenced sub-schemas by inheritance from the following object classes of the basic schema.

- 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 (Object)
- AU_Punktobjekt (PointObject)
- AU_Punkthaufenobjekt (MultiPointObject)
- AU_Linienobjekt (LineObject)
- AU_KontinuierlichesLinienobjekt (ContinuousLineObject)
- AU_FlaechenObjekt (SurfaceObject)
- AU_Punktobjekt_3D (PointObject)
- AU_Umringobjekt_3D (RingObject)
- AU_Punkthaufenobjekt_3D (MultiPointObject)
- AU_Mehrfachflaechenobjekt_3D (MultipleSurfacesObject)
- AU_MehrfachlinienObjekt_3D (MultipleLineObject)
- AU_Geometrieobjekt_3D (GeometryObject)
- AU_KoerperObjekt_3D (SolidObject)
- AU_TrianguliertesOberflaechenObjekt_3D (TriangulatedSurfaceObject)
- AD_PunktCoverage (PointCoverage)
- AD_GitterCoverage (RectifiedGridCoverage)

For presentation objects the following object classes of the basic schema can be used directly or instantiated.

- AP_PPO
- AP_PTO
- AP_LTO
- AP_LPO
- AP_FPO

- AP_Presentation
- AP_KPO_3D

Alternatively, other instantiable, technical object classes can be derived by inheritance from these object classes of the basic schema.

3.3.2 Attributes

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

Attributes of type Date, Time or DateTime are modelled according to ISO 8601, chapter 5.4 in connection with 5.3.3. The variant with a separator has been chosen. Accuracy of the time is the full second, time zone is always UTC (Universal Time Coordinated, also known as Greenwich Mean Time, abbreviation: Z(T)). Example: 2004-04-01T17:06:31Z

3.3.3 Associations

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

- According to the ISO *General Feature Model*, features 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 overpass or underpass relations, because such relations always only contain the duple association between the participating objects.

– Map geometry

The relation between map geometry objects (=generalised geometry, to 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 should point to a technical data object being managed in an external technical data system, this can alternatively be described through the *verweistAufExternes (indicatesExternal)* attribute. Thematic data connections are structured according to section 3.3.9.

The construction of the thematic data connection is expedient in order to consider the existence of thematic data inventories during the use and management of ALKIS.

Depiction relation

Presentation objects serve to depict objects of the primary database data. This relation is represented by the *dientZurDarstellungVon (isUsedtoDepict)* reference between the presentation object and other objects.

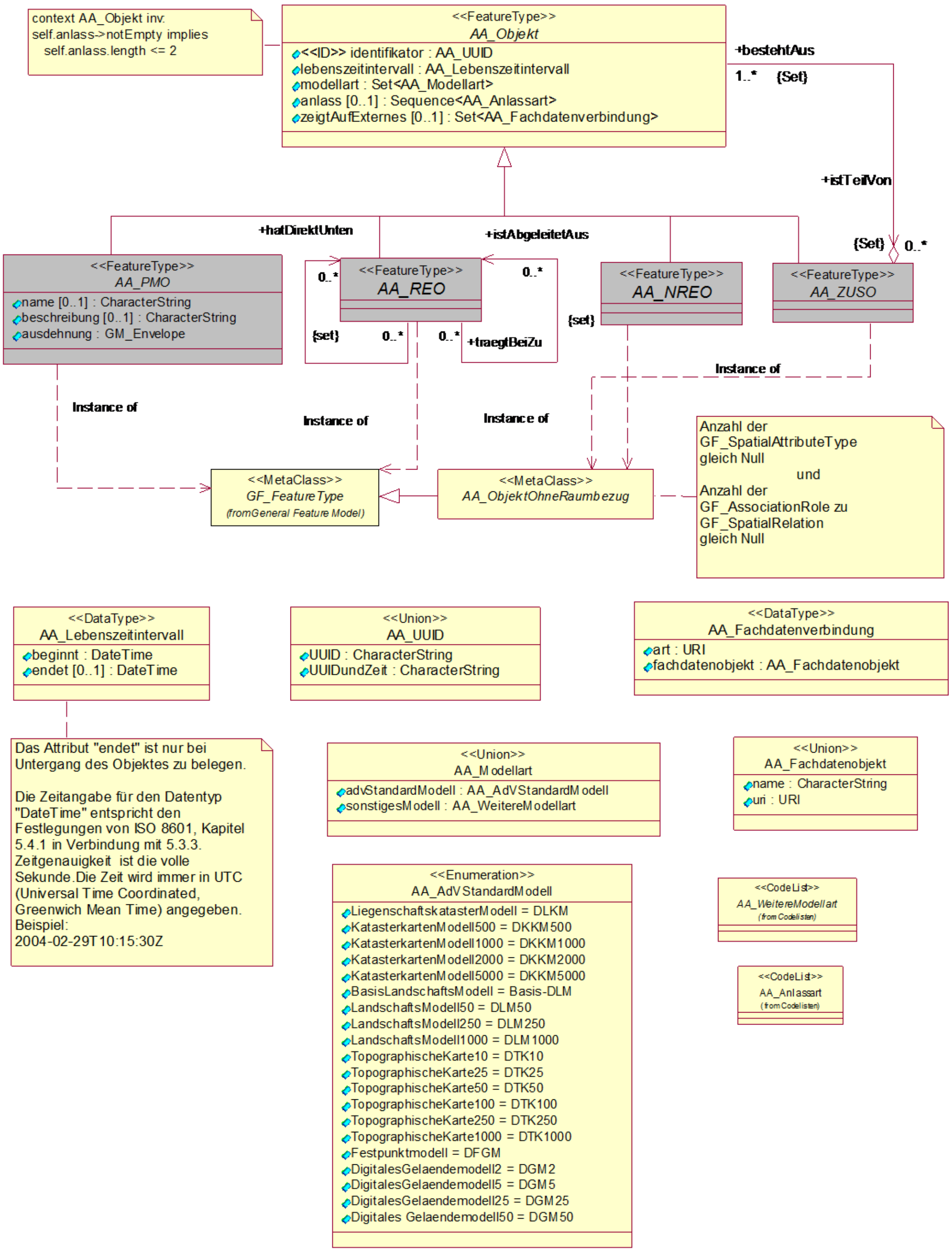

context AA_Objekt inv:
self.anlass->notEmpty implies
self.anlass.length <= 2
<<FeatureType>>
AA_Objekt
<<ID>> identifikator : AA_UUID
lebenszeitintervall : AA_Lebenszeitintervall
modellart : Set<AA_Modellart>
anlass [0..1] : Sequence<AA_Anlassart>
zeigtAufExternes [0..1] : Set<AA_Fachdatenverbindung>
+bestehtAus
1..*
{Set}
+istTeilVon
{Set}
0..*
+hatDirektUnten
+istAbgeleitetAus
<<FeatureType>>
AA_PMO
name [0..1] : CharacterString
beschreibung [0..1] : CharacterString
ausdehnung : GM_Envelope
<<FeatureType>>
AA_REO
0..*
{set}
0..*
0..*
0..*
+traegtBeiZu
{set}
<<FeatureType>>
AA_NREO
<<FeatureType>>
AA_ZUSO
Instance of
Instance of
Instance of
Instance of
<<MetaClass>>
GF_FeatureType
(fromGeneral Feature Model)
<<MetaClass>>
AA_ObjektOhneRaumbezug
Anzahl der
GF_SpatialAttributeType
gleich Null
und
Anzahl der
GF_AssociationRole zu
GF_SpatialRelation
gleich Null
<<DataType>>
AA_Lebenszeitintervall
beginnt : DateTime
endet [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
<<Enumeration>>
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
<<CodeList>>
AA_WeitereModellart
(from Codelisten)
<<CodeList>>
AA_Anlassart
(from Codelisten)

Figure 8: Modelling of AAA 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; of these, solely the following constructs are used for AFIS, ALKIS and ATKIS, in order to reduce complexity.

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_MultiPoint GM_MultiCurve GM_MultiSurface	TS_PointComponent TS_CurveComponent TS_SurfaceComponent TS_Face	TP_Complex

The representation of 3D geometries is also based on ISO 19107. The geometrical and topological primitives present in the AAA basic schema are extended with those displayed in the following table.

Geometric Objects (GM_Object)			**Topological Objects (TP_Object)**	
Geometric Primitives	**Geometric Complexes**	**Geometric Aggregates**	**Topological Primitives**	**Topological Complexes**
GM_Solid GM_SurfaceBoundary GM_TriangulatedSurface GM_OrientableSurface	GM_CompositeSolid		TS_Solid TS_Feature TS_Theme	

The geometric and topological objects are described as UML classes. The standard also contains spatial operations which the geometric and topological objects *(GM_Object* or *TP_Object)* use as parameters (create, delete, change, spatial evaluations ...). The defined classes have no direct application, i.e. they are not instantiable. 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 directly used, however, in this application for simplification purposes.

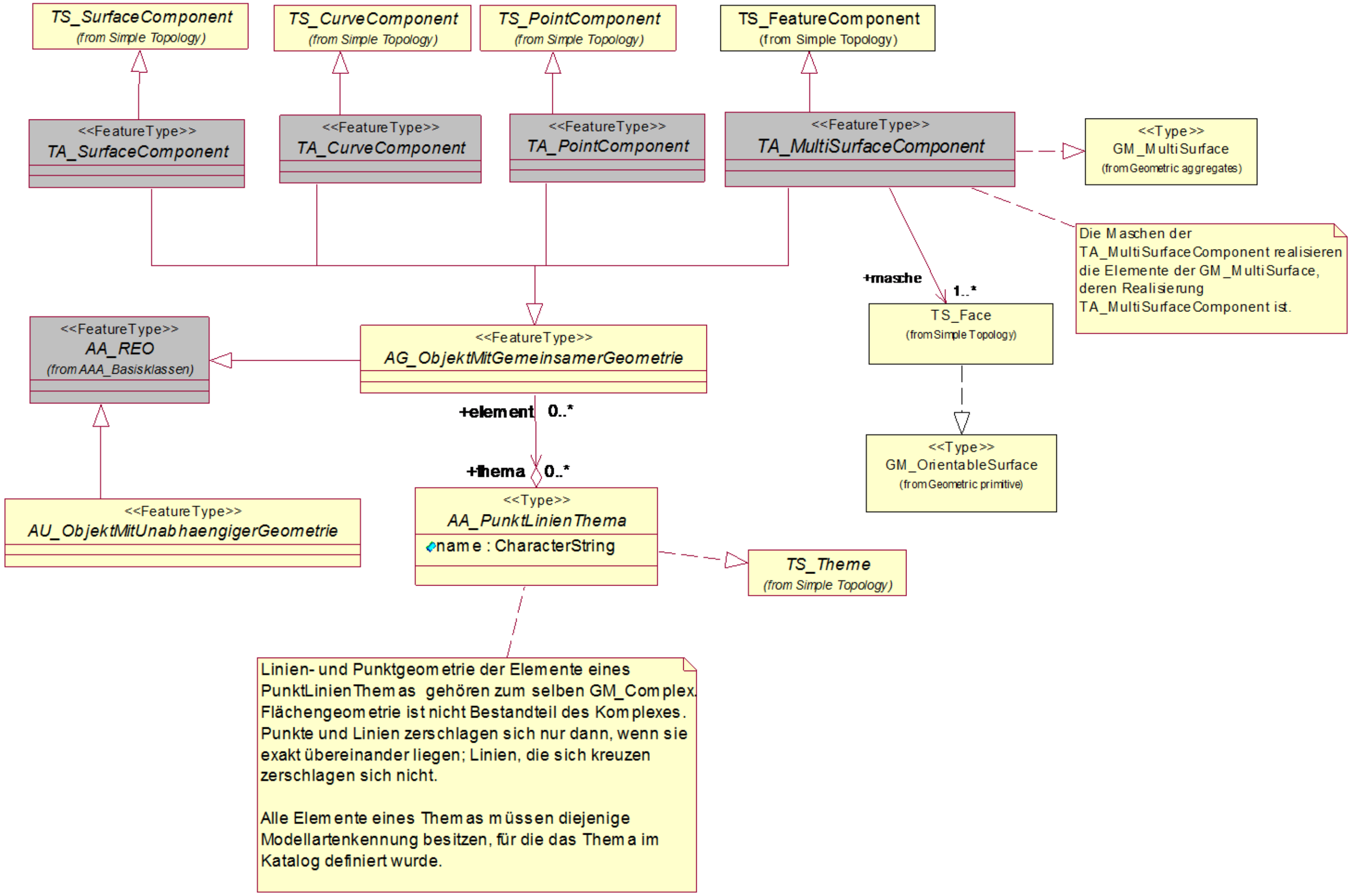
The geometric primitives usually appear as attributes of objects; this does not mean, however, that the geometry is always redundant in principle. The common AFIS-ALKIS-ATKIS application schema has the following options for linking the spatial reference:

- The formation of node-shaped, edge-shaped and face-shaped objects with "simple topology". Additionally, 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.
- Formation of point, line, surface and volume objects, which share lines and points.
- Formation of point, line, surface and volume objects with "independent" geometry.
- Formation of topological and geometrical "topics" that allow the selective combination of feature types into complexes, in order to express geometric identities and/or topological correlations.
- TriangulatedSurface (Foundation of 3D-DGM)

- A triangulated surface results from the subdivision of surfaces into triangles through triangulation, using algorithms such as Delaunay Triangulation.
- For example the geometry of a Tin relief is defined by the GML geometry type gml.TriangulatedSurface. In thematic schemas, either the geometry type gml:TriangulatedSurface or its subclass gml:tin can be used.

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

The necessary enhancements and restrictions of the ISO *Spatial Schema* are summarised in the following diagrams:

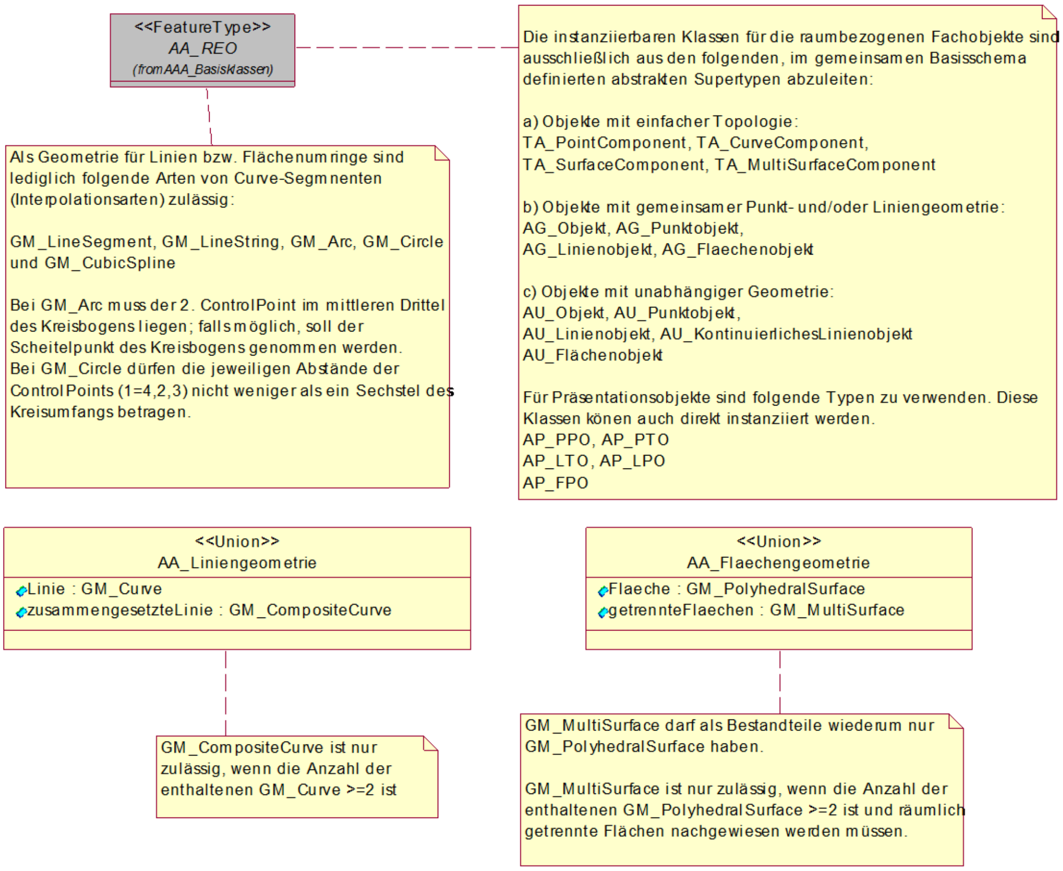


Die Maschen der TA_MultiSurfaceComponent realisieren die Elemente der GM_MultiSurface, deren Realisierung TA_MultiSurfaceComponent ist.

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: Summary of the enhancements to the standardised Spatial Schema required by AFIS-ALKIS-ATKIS



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 instanziiert werden.
AP_PPO, AP_PTO
AP_LTO, AP_LPO
AP_FPO

Als Geometrie für Linien bzw. Flächenumringe sind lediglich folgende Arten von Curve-Segmnenten (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: Geometry and instantiable classes

Regarding the use of geometries for spatially-referenced objects, the following restrictions in the description of the classes are specified:

The choice data type AA_Flaechengeometrie (SurfaceGeometry) allows the alternative modelling of surface objects through a surface or a number of surfaces.

GM_MultiSurface is only permitted if the number of GM_PolyhedralSurface contained is >=2 and if spatially separated surfaces have to be captured Surfaces that are not spatially separated are always to be represented by 1 surface (GM_PolyhedralSurface) unless the capture of very large surfaces necessitates a GM_CompositeSurface.

The drop-down data type AA_Liniengeometrie allows line objects to be geometrically modelled either by a single line or by several successive lines. GM_CompositeCurve is only permitted if the number of GM_Curve contained is >=2.

The instantiable classes for the spatially-referenced features are to be derived exclusively from the following abstract supertypes defined in the common basic schema:

Objects with simple topology

- TA_PointComponent,
- TA_CurveComponent
- TA_SurfaceComponent,
- TA_MultiSurfaceComponent

Objects with common point and/or line geometry

- AG_Objekt (Object)
- AG_Punktobjekt (PointObject)
- AG_Linienobjekt (LineObject)
- AG_Flaechenobjekt (SurfaceObject)

Objects with independent geometry

- AU_Objekt (Object)
- AU_Punktobjekt (PointObject)
- AU_PunkthaufenObjekt (MultiPointObject)
- AU_Linienobjekt (LineObject)
- AU_KontinuierlichesLinienobjekt (ContinuousLineObject)
- AU_Flächenobjekt (SurfaceObject)
- AU_TrianguliertesOberfaechenObjekt_3D (TriangulatedSurfaceObject)
- AU_MehrfachLinienObjekt_3D (MultipleLineObject)
- AU_MehrfachFlaechenObjekt_3D (MultipleSurfacesObject)
- AU_UmringObjekt_3D (RingObject)
- AU_Punktobjekt3D (PointObject)
- AU_PunkthaufenObjekt_3D (MultiPointObject)
- AU_KoerperObjekt_3D (SolidObject)
- AU_GeometrieObjekt_3D (GeometryObject)

For presentation objects the following types are to be used. These classes can also be directly instantiated:

- AP_PPO
- AP_PTO
- AP_KPO_3D
- AP_LTO
- AP_LPO
- AP_FPO

As a geometry for lines or surface-rings only the following types of curve segments (interpolation types) are permitted:

- GM_LineSegment
- GM_LineString
- GM_Arc
- GM_Circle
- GM_CubicSpline.

With GM_Arc, the 2nd ControlPoint must lie in the middle third of the arc; if possible, the apex of the arc should be taken. With GM_Circle, each of the intervals between the ControlPoints (1=4,2,3) may not amount to less than a sixth of the circumference.

3.3.4.2 Features with simple topology

ISO 19107 *Spatial Schema* provides the *Simple Topology* schema as a module for an application schema. Based on this, features are made available 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 provide the common AFIS-ALKIS-ATKIS object properties (identifier, lifetime interval, cause of change) and also the option of linking various feature types geometrically and topologically via the construct of the "PointLineTheme". The TA-classes may belong to a topological theme and one or more PointLineThemes at the same time. The TA_MultiSurfaceComponent class has been defined divergently to the TA_SurfaceComponent class to enable the referenced faces *(TS_Face)* to implement 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.

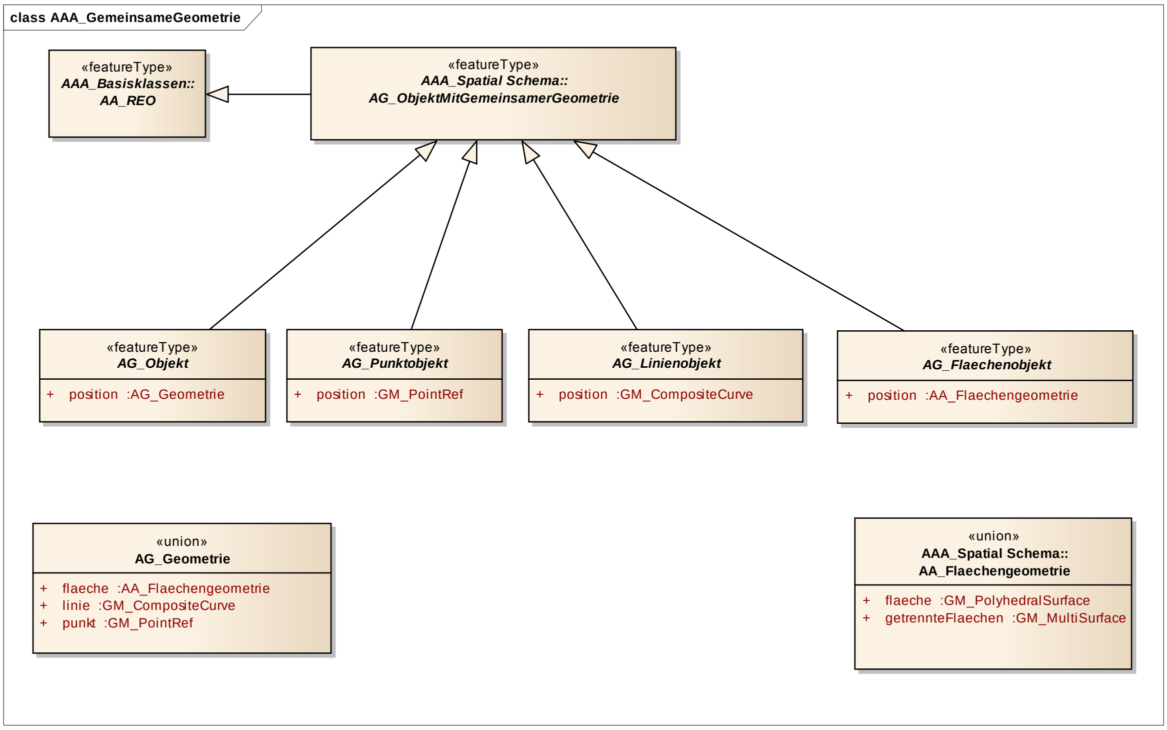


Figure 11: Objects with jointly used geometry

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 jointly 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-referenced feature types with common geometry.

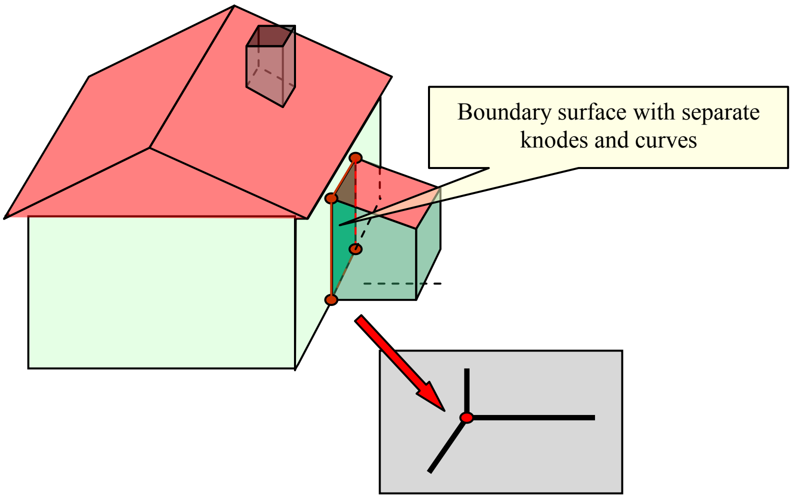


Figure 10: 3D-features with yointly used geometry

3.3.4.4 Features with independent geometry

Two packages describe features with independent geometry:

- The "AAA_IndependentGeometry" package provides six basic classes of features, whose geometry consist of independent points, lines andsurfaces.
- The AAA_Independent Geometry 3D package provides basic classes of 3D features, whose geometries are independent of one another.

These basic classes are to be used as a basis for spatially-referenced feature types with independent geometry (e.g. presentation objects).

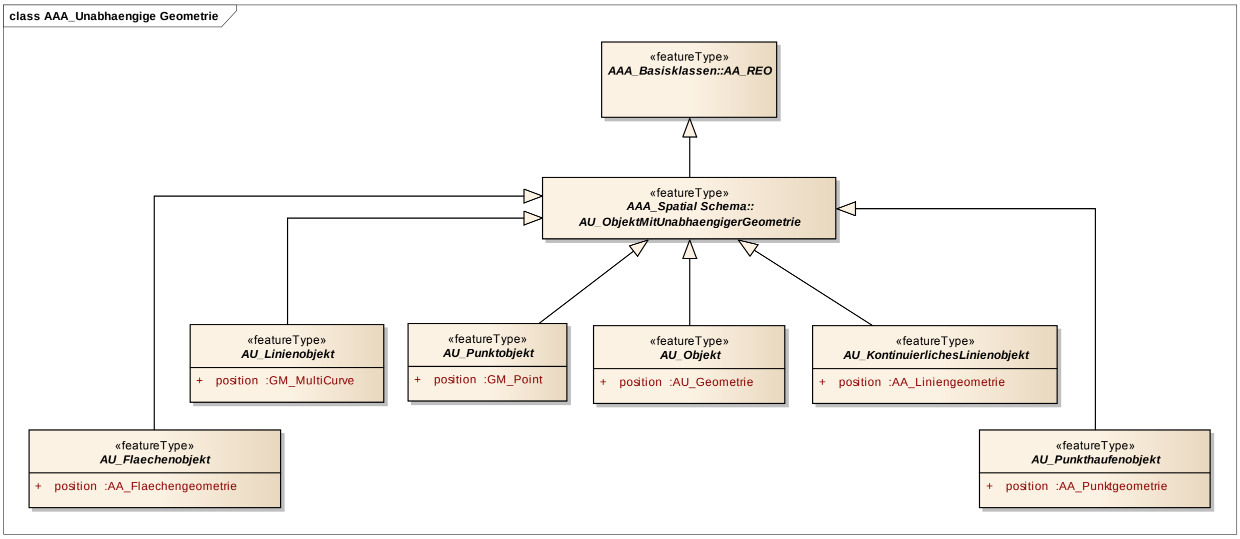


Figure 13: 2D objects with independent geometry

AU_Flaechenobjekt (SurfaceObject) is used for features that are described geometrically by surfaces.

An AU_Linienobjekt (LineObject) is described geometrically by a set of lines (use case: e.g. rock signature).

AU_Punktobjekt (PointObject) contains a feature that is represented geometrically by a single point.

The choice data type "AU_Objekt" allows subclasses to be formed whose concrete type of geometry is first defined at the instantiation level. Variously dimensional geometries are available: point, line, surface and composite line. This makes it possible, for example, to form a "Tower" object type that, depending on its extent in the real world, has a point or surface geometry.

Features that are described geometrically by connected lines inherit from "AU_KontinuierlichesLinienobjekt" (ContinuousLineObject). These objects must not intersect or overlap. Use case: e.g. pipeline.

"AU_Punkthaufenobjekt" (MultiPointObject) is used for an object that is represented geometrically by a point or a multi point. The choice data type "AA_Punktgeometrie" (PointGeometry) allows point objects to be modelled either with one or several point geometries. Its use only seems to make sense for objects with independent geometry. (e.g. presentation objects with symbol clusters with individual geometry).

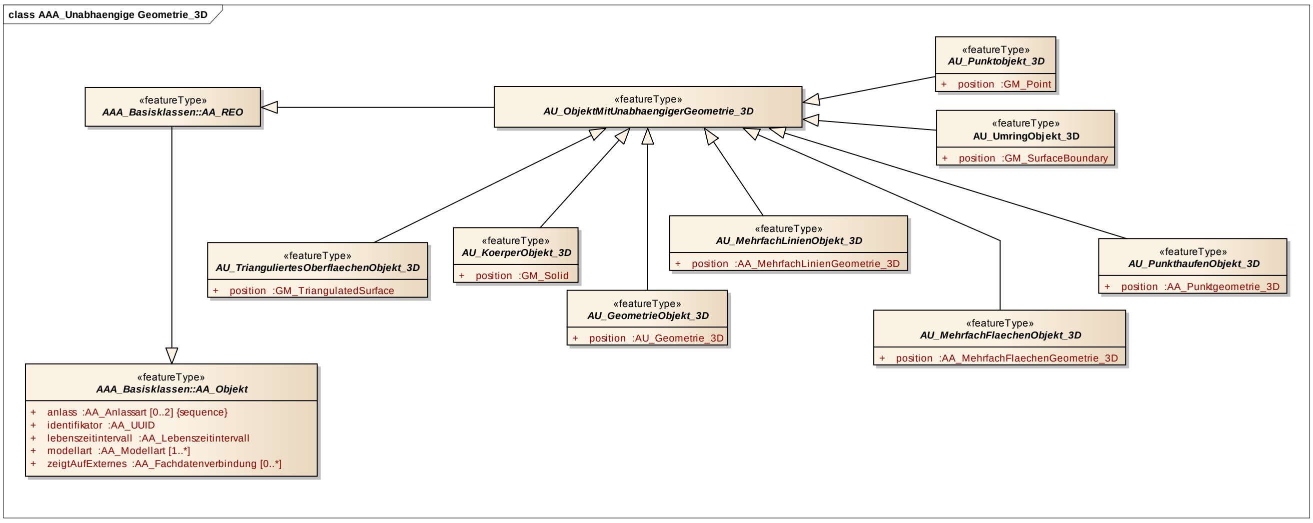

class AAA_Unabhaengige Geometrie_3D
«featureType»
AAA_Basisklassen::AA_REO
«featureType»
AU_ObjektMitUnabhaengigerGeometrie_3D
«featureType»
AU_Punktobjekt_3D
+ position :GM_Point
«featureType»
AU_UmringObjekt_3D
+ position :GM_SurfaceBoundary
«featureType»
AU_TrianguliertesOberflaechenObjekt_3D
+ position :GM_TriangulatedSurface
«featureType»
AU_KoerperObjekt_3D
+ position :GM_Solid
«featureType»
AU_MehrfachLinienObjekt_3D
+ position :AA_MehrfachLinienGeometrie_3D
«featureType»
AU_PunkthaufenObjekt_3D
+ position :AA_Punktgeometrie_3D
«featureType»
AU_GeometrieObjekt_3D
+ position :AU_Geometrie_3D
«featureType»
AU_MehrfachFlaechenObjekt_3D
+ position :AA_MehrfachFlaechenGeometrie_3D
«featureType»
AAA_Basisklassen::AA_Objekt
+ anlass :AA_Anlassart [0..2] {sequence}
+ identifikator :AA_UUID
+ lebenszeitintervall :AA_Lebenszeitintervall
+ modellart :AA_Modellart [1..*]
+ zeigtAufExternes :AA_Fachdatenverbindung [0..*]

Figure 14: 3D objects with independent geometry

Overlapping of 3D objects with independent geometries is allowed. Modelling is much simpler as there are no restrictions with the common usage of different geometrical and topological primitives. This type of geometry is sufficient for visualisation purposes.

When certifying geospatial reference data (block model) a redundancy free geometry is recommended, however, or the overlapping of geometries should be avoided.

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

"AU_ObjektMitUnabhaengigerGeometrie_3D" (ObjectWithIndependentGeometry) is the super class to the eight classes with independent 3D geometry. An "AU_ObjektMitUnabhaengigerGeometrie_3D" (ObjectWithIndependentGeometry) is a spatially-referenced elementary object for technical 3D applications whose subclasses may not share geometries at the level of the instance. The class is not directly instantiable.

1. AU_Punktobjekt_3D (Pointobject) is a 3D feature that is represented geometrically by a single 3D point.

2. AU_PunkthaufenObjekt_3D (MultiPointobject) is a 3D feature that is represented geometrically by a 3D multi point.

3. AU_MehrfachLinienObjekt_3D (MultipleLineObject) is a 3D feature that is described geometrically by 3D lines. Multiple, separated 3D lines are permitted.

4. AU_MehrfachFlaechenObjekt_3D (MultipleSurfacesObject) is a 3D feature that is described geometrically by 3D surfaces. Multiple, separated 3D surfaces are permitted.

5. AU_TrianguliertesOberflaechenObjekt_3D (TriangulatedSurfaceObject) is a 3D feature that is described geometrically by spatially related 3D surfaces which define a triangulated surface (TIN) (e.g. a terrain surface),

6. AU_KoerperObjekt_3D (SolidObject) is a 3D feature that is described geometrically by 3D solids.

7. AU_UmringObjekt_3D (RingObject) is a 3D feature that is described by a 3D ring polygon and can have further 3D ring polygons for enclaves. Every part of the geometry must be a closed ring.

8. AU_GeometrieObjekt_3D (GeometryObject) is a 3D feature that enables subclasses to be formed whose concrete type of 3D geometry is first defined at the instantiation level e.g. dependent on the detail level (Level of Detail) (e.g. walls that can be formed by 3D surfaces or in more detail by 3D solids).

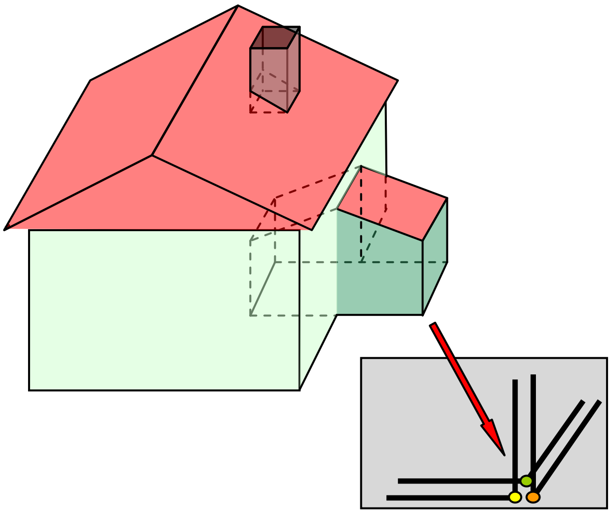


Figure 11: Features with independent 3D geometry

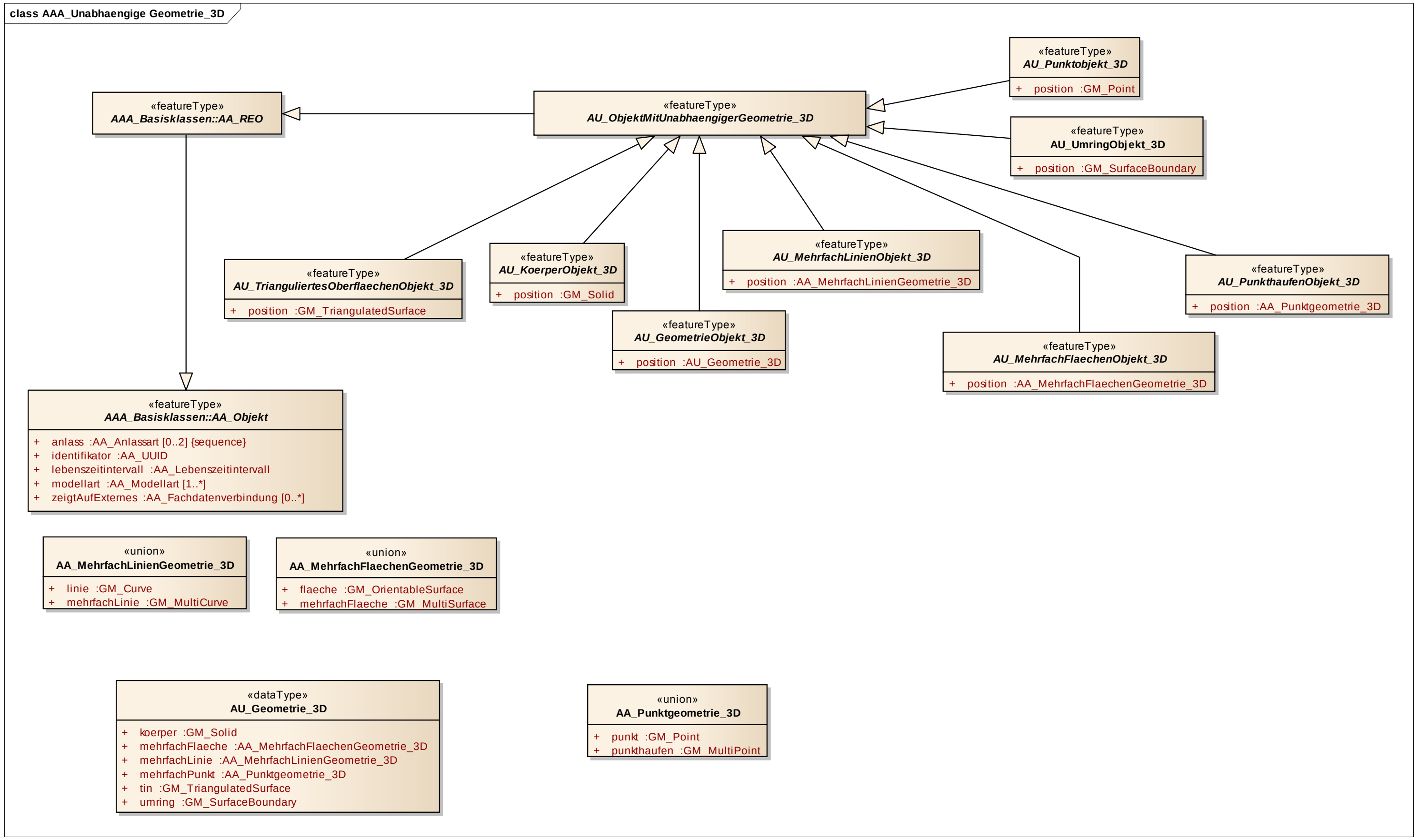
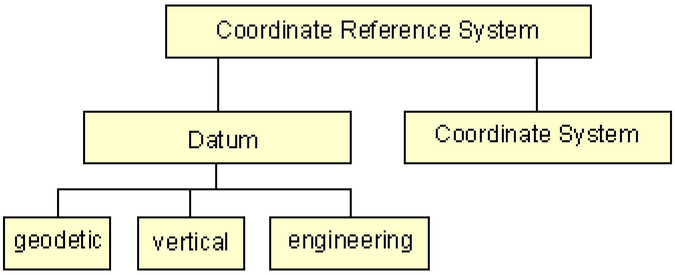

class AAA_Unabhaengige Geometrie_3D
«featureType»
AU_Punktobjekt_3D
+ position :GM_Point
«featureType»
AAA_Basisklassen::AA_REO
«featureType»
AU_ObjektMitUnabhaengigerGeometrie_3D
«featureType»
AU_UmringObjekt_3D
+ position :GM_SurfaceBoundary
«featureType»
AU_TrianguliertesOberflaechenObjekt_3D
+ position :GM_TriangulatedSurface
«featureType»
AU_KoerperObjekt_3D
+ position :GM_Solid
«featureType»
AU_MehrfachLinienObjekt_3D
+ position :AA_MehrfachLinienGeometrie_3D
«featureType»
AU_PunkthaufenObjekt_3D
+ position :AA_Punktgeometrie_3D
«featureType»
AU_GeometrieObjekt_3D
+ position :AU_Geometrie_3D
«featureType»
AU_MehrfachFlaechenObjekt_3D
+ position :AA_MehrfachFlaechenGeometrie_3D
«featureType»
AAA_Basisklassen::AA_Objekt
+ anlass :AA_Anlassart [0..2] {sequence}
+ identifikator :AA_UUID
+ lebenszeitintervall :AA_Lebenszeitintervall
+ modellart :AA_Modellart [1..*]
+ zeigtAufExternes :AA_Fachdatenverbindung [0..*]
«union»
AA_MehrfachLinienGeometrie_3D
+ linie :GM_Curve
+ mehrfachLinie :GM_MultiCurve
«union»
AA_MehrfachFlaechenGeometrie_3D
+ flaeche :GM_OrientableSurface
+ mehrfachFlaeche :GM_MultiSurface
«dataType»
AU_Geometrie_3D
+ koerper :GM_Solid
+ mehrfachFlaeche :AA_MehrfachFlaechenGeometrie_3D
+ mehrfachLinie :AA_MehrfachLinienGeometrie_3D
+ mehrfachPunkt :AA_Punktgeometrie_3D
+ tin :GM_TriangulatedSurface
+ umring :GM_SurfaceBoundary
«union»
AA_Punktgeometrie_3D
+ punkt :GM_Point
+ punkthaufen :GM_MultiPoint

Figure 12: Features with independent 3D geometry

3.3.4.5 Spatial reference system, coordinates

For each geometry the associated **coordinate reference system (CRS)** can be specified and stored in AFIS-ALKIS-ATKIS.



According to ISO 19111 *(Spatial Referencing by Coordinates)* a coordinate reference system consists of two components, the "datum" and the "coordinate system" (see sketch).

The **datum** is the physical part of a CRS that, using the definition of an origin, orientation of the coordinate axis and a scale, defines a reference to earth. A datum can be a geodetic datum, a vertical datum or an engineering (local) datum. Examples for a geodetic datum is the German Hauptdreiecksnetz (DHDN), also called "Potsdam-Datum", or the European Terrestrial Reference System (ETRS 89).

The **coordinate system** ist the mathematical part of a CRS that uses rules to define how to apply coordinates to a geometry, for example a geodetic fixed point. The coordinates of a geometry can be Cartesian coordinates (X,Y,Z), ellopsoidal corrdinates (latitude, longitute, and ellipsoidal height) or projected coordinates (Gauß-Krueger-projection, UTM).

Besides the CRS for 2D locational and 3D spatial information there are private coordinate reference systems for managing height or height coordinates (e.g. NN height). The coordinate reference systems commonly used in Germany are listed in the section "Coordinate reference systems and Units of Measure" 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 a combined CRS can also be introduced. Nevertheless, for features of feature type "AX_PointLocation" combined coordinate reference systems are **not** allowed in AFIS-ALKIS-ATKIS.

Because the formerALK and ATKIS systems did not contain details on the CRS the data depends on, a task of the migration concept was to define the relevant specifications.

3.3.5 Portrayal, Presentation objects

On account of their generally applicable properties/features, the presentation objects are described in the AAA basic schema.

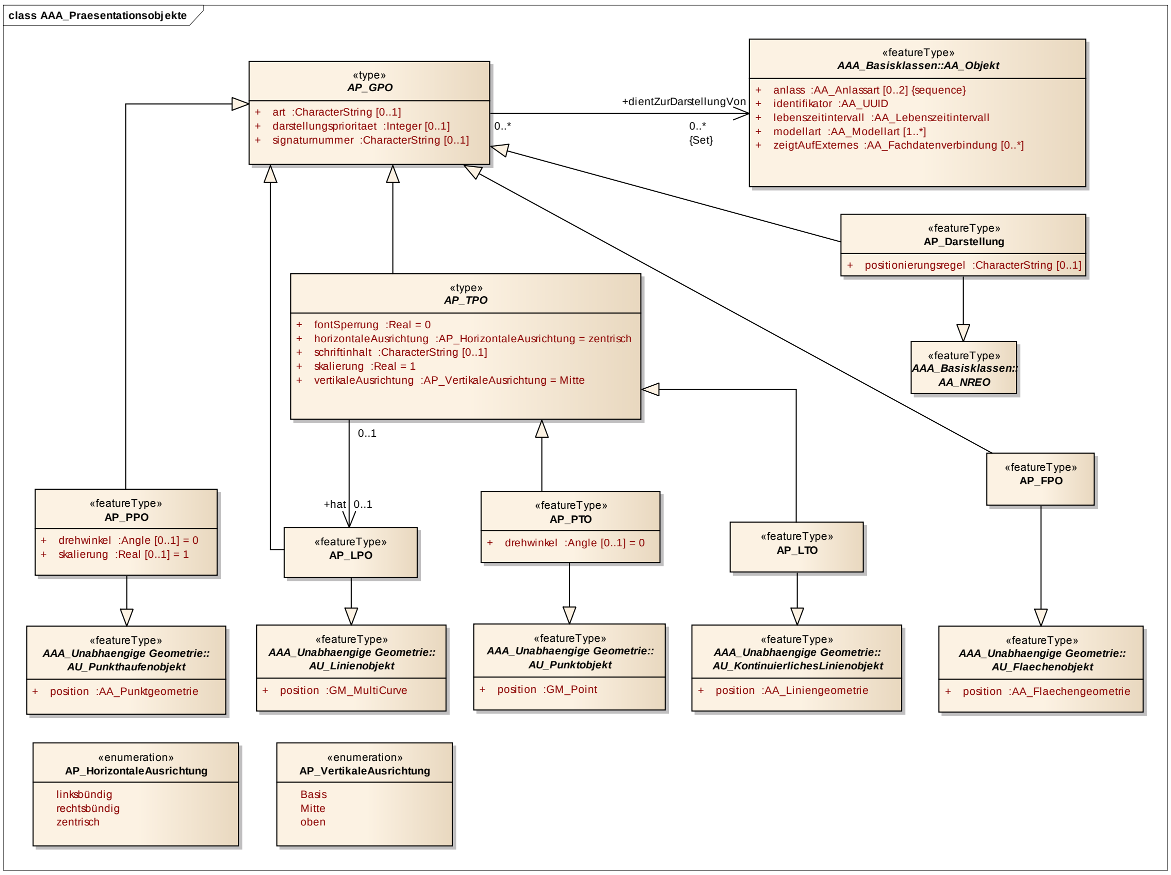

class AAA_Praesentationsobjekte
«type»
AP_GPO
+ art :CharacterString [0..1]
+ darstellungsprioritaet :Integer [0..1]
+ signaturnummer :CharacterString [0..1]
+dientZurDarstellungVon
0..*
0..*
{Set}
«featureType»
AAA_Basisklassen::AA_Objekt
+ anlass :AA_Anlassart [0..2] {sequence}
+ identifikator :AA_UUID
+ lebenszeitintervall :AA_Lebenszeitintervall
+ modellart :AA_Modellart [1..*]
+ zeigtAufExternes :AA_Fachdatenverbindung [0..*]
«featureType»
AP_Darstellung
+ positionierungsregel :CharacterString [0..1]
«featureType»
AAA_Basisklassen::
AA_NREO
«type»
AP_TPO
+ fontSperrung :Real = 0
+ horizontaleAusrichtung :AP_HorizontaleAusrichtung = zentrisch
+ schriftinhalt :CharacterString [0..1]
+ skalierung :Real = 1
+ vertikaleAusrichtung :AP_VertikaleAusrichtung = Mitte
0..1
+hat 0..1
«featureType»
AP_PPO
+ drehwinkel :Angle [0..1] = 0
+ skalierung :Real [0..1] = 1
«featureType»
AP_LPO
«featureType»
AP_PTO
+ drehwinkel :Angle [0..1] = 0
«featureType»
AP_LTO
«featureType»
AP_FPO
«featureType»
AAA_Unabhaengige Geometrie::
AU_Punkthaufenobjekt
+ position :AA_Punktgeometrie
«featureType»
AAA_Unabhaengige Geometrie::
AU_Linienobjekt
+ position :GM_MultiCurve
«featureType»
AAA_Unabhaengige Geometrie::
AU_Punktobjekt
+ position :GM_Point
«featureType»
AAA_Unabhaengige Geometrie::
AU_KontinuierlichesLinienobjekt
+ position :AA_Liniengeometrie
«featureType»
AAA_Unabhaengige Geometrie::
AU_Flaechenobjekt
+ position :AA_Flaechengeometrie
«enumeration»
AP_HorizontaleAusrichtung
linksbündig
rechtsbündig
zentrisch
«enumeration»
AP_VertikaleAusrichtung
Basis
Mitte
oben

Figure 13: Presentation objects

Presentation objects contain the signature number and other properties to control the presentation, such as portrayal priority and type. In ALKIS, presentation objects must be connected to the corresponding feature by a "dientzurDarstellungVon" (servesPresentationOf) relation. There is no such rule in AKTIS which means the "free PresentationObjects" can exist. Presentation of objects in graphical as well as non-graphical output is handled as illustrated in the following figure as follows:

Presentation in the map

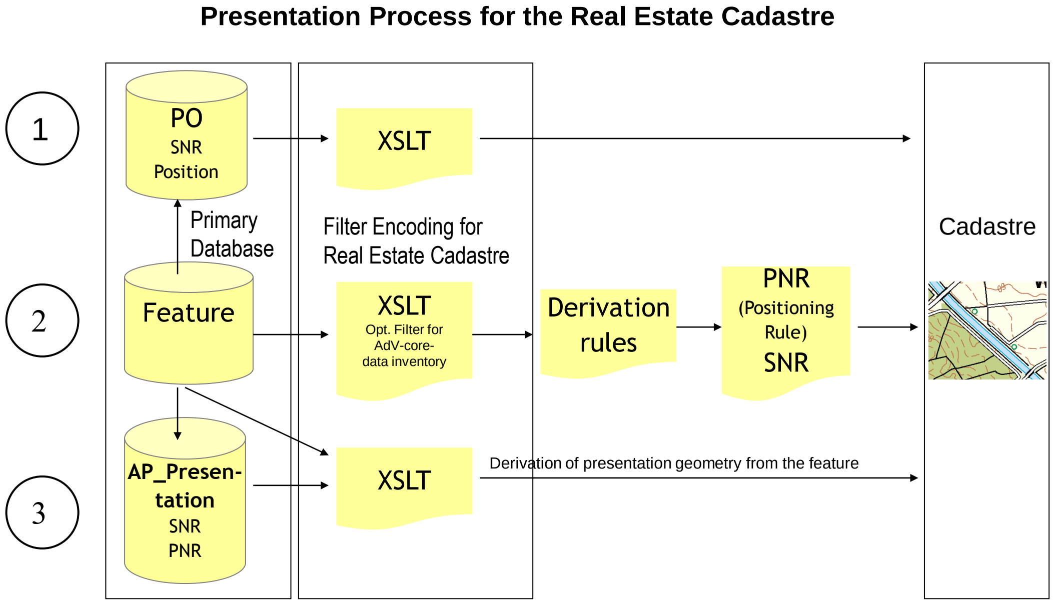


Figure 14: Presentation flow for the real estate map

1. Presentation objects in primary database

   Presentation objects will be built in form of text, line, symbol and area, for all portrayals that cannot be automatically created and positioned. The specific portrayal number, which represents a derivation rule, as well as the positioning number (PNR), which stands for a specific positioning rule can be optionally stored in the presentation object. Presentation objects are also to be generated, if the output must deviate from the defined standard presentation, stipulated in the portrayal catalogue (e.g. different font height for plot number).

2. Presentation using derivation rule and positioning rule

   Portrayals of a feature in form of text, line, symbol and area will be placed at a defined position (standard position) by using the filter encoding and a concrete derivation rule that leads to a specific portrayal number and the position rule that activates a specific positioning number. In this case a presentation object is not created in the data in primary database. The technical information to be presented is determined from specified attribute type of the feature type. This is the standard case that, due to performance reasons, is not always the most efficient.

3. Presentation using stored derivation rule and positioning rule

   In order to increase the presentation performance of the standard case, the concrete portrayal number as well as the positioning that will be used to present a feature are stored at a certain time (updating, first insert) under the referenced presentation object AP_Darstellung (AP_Presentation) as NREO. The advantage over forming the presentation object (case 1) is avoidance of redundancies in the geometry, as with AP_Darstellung (AP_Presentation) the presentation geometry is derived from the respective feature. The presentation will be executed as quickly as possible, by using the filter encoding in connection with the derivation of the presentation geometry from the feature and the stored rules, i.e. portrayal number, positioning number. In ALKIS the feature type AP_Darstellung (AP_Presentation) is used to make changes such as the following in a real estate map:

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

Presentation of a real estate description

The presentation of the information concerning the cadastral content, e.g. plot certification and owner certification, occurs automatically, on the fly, by using Filter Encoding in order to create the corresponding output data. The output data is presented by using a specific derivation rule. The corresponding text positions can be taken from the corresponding symbols. The non-spatial feature type AP_Darstellung (AP_Presentation) is not used in this context. Similarly, no presentation objects are created in the primary data base.

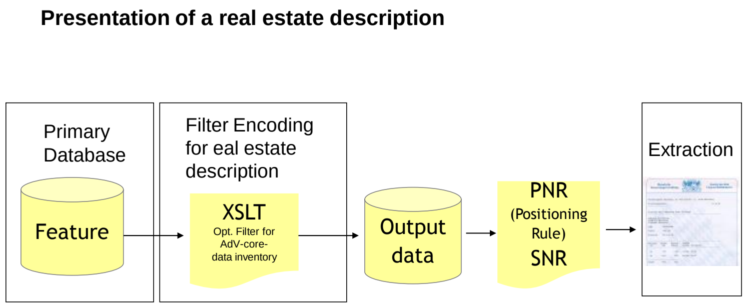

Presentation of a real estate description
Primary Database
Feature
Filter Encoding for eal estate description
XSLT
Opt. Filter for AdV-core-data inventory
Output data
PNR
(Positioning Rule)
SNR
Extraction

Figure 15: Presentation of a real estate description

Creation of presentation objects and AP_Darstellung (AP_Presentation) for data in the primary data base

In order to ensure an efficient presentation of the features in a cadastral extract, appropriate presentation rules must be defined at the time when the data is being collected or updated. In doing so three cases are identified, illustrated in the following figure.

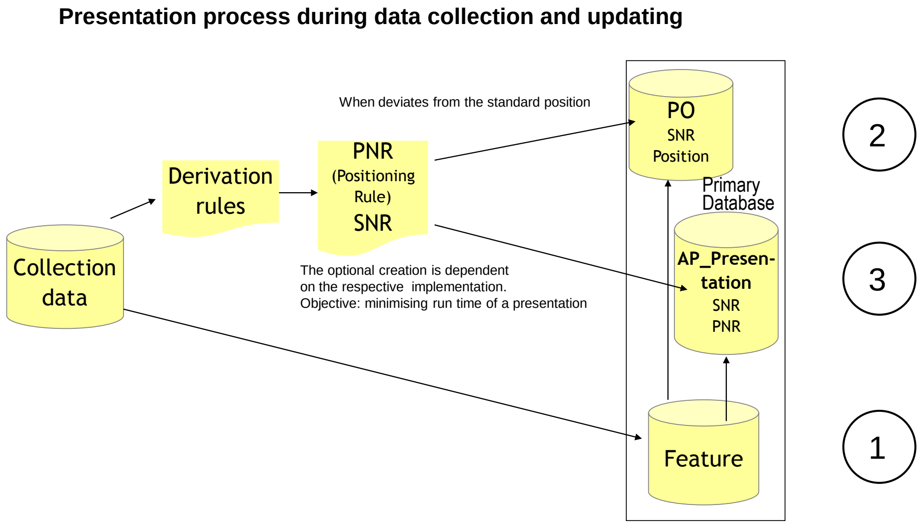

Presentation process during data collection and updating
When deviates from the standard position
Derivation rules
PNR
(Positioning Rule)
SNR
Collection data
The optional creation is dependent on the respective implementation.
Objective: minimising run time of a presentation
PO
SNR
Position
Primary Database
AP_Presen-tation
SNR
PNR
Feature
2
3
1

Figure 16: Presentation process during data collection and updating

1. No definition of presentation rules
   Collected or updated data does not need predefined rules to assign a concrete portrayal number or positioning number in order to realize a rapid presentation.

The information required for a presentation can be generated directly on the fly from the 3A data model as well as the ALKIS portrayal catalogue.

2. Storing presentation objects in the primary database
   During the collecting or updating process a presentation object is created to present a concrete portrayal of a feature, as the portrayals for example, could not be automatically generated and placed for a specific map scale. In this case the information about the geometry, optionally a portrayal number or optionally a positioning number, are stored in the object.
3. Definition of presentation rules
   In order to minimize the runtime of a presentation a non-spatial feature type AP_Presentation for a specific feature can be generated during the data collection or updating process. In this feature type a concrete position number is stored, e.g. an arrangement of symbols in an area. The geometry for the presentation can be derived from the corresponding feature on the fly using appropriate functions.

3D Presentation objects

The AAA_Praesentationsobjekte_3D (PresentationObject) concretises features with AAA_Unabhaengige Geometrie 3D (Independent Geometry) for presentation purposes. The corresponding feature can be immediately instantiated.

The AP_KPO_3D 3D presentation object is used for 3D symbols whose 3D geometry is stored in an external data format and referenced via an URI. AP_KPO_3D is derived from AU_Punktobject_3D (PointObject) and its 3D point geometry positions the symbol. Using a transformation matrix a location independent 3D geometry in the external data format is transformed to the spatial reference of the presentation object AP_KPO_3D. The presentation objects are to be defined, like other objects in the feature type catalogue, in connection with the respective portrayal catalogue or 3D symbol library.

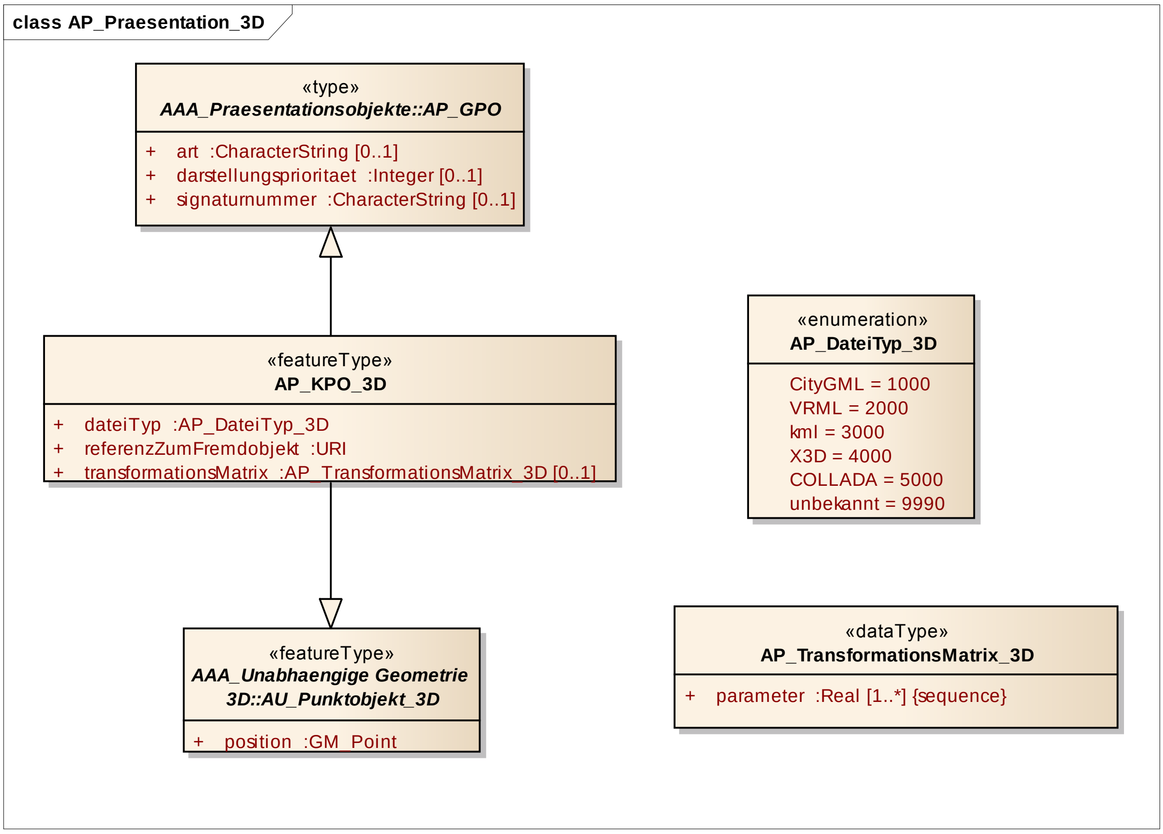

class AP_Praesentation_3D
«type»
AAA_Praesentationsobjekte::AP_GPO
+ art :CharacterString [0..1]
+ darstellungsprioritaet :Integer [0..1]
+ signaturnummer :CharacterString [0..1]
«featureType»
AP_KPO_3D
+ dateiTyp :AP_DateiTyp_3D
+ referenzZumFremdobjekt :URI
+ transformationsMatrix :AP_TransformationsMatrix_3D [0..1]
«featureType»
AAA_Unabhaengige Geometrie 3D::AU_Punktobjekt_3D
+ position :GM_Point
«enumeration»
AP_DateiTyp_3D
CityGML = 1000
VRML = 2000
kml = 3000
X3D = 4000
COLLADA = 5000
unbekannt = 9990
«dataType»
AP_TransformationsMatrix_3D
+ parameter :Real [1..*] {sequence}

Figure 17: PresentationObject_3D

Regulations for countrywide, redundancy free allocation of Federal State specific portrayal numbers

In addition to many other advantage, the AFIS-ALKIS-ATKIS data model brings the users a countrywide unification of the data inventories and their presentation. The following text is contained in part A of the ALKIS portrayal catalogue:

"The ALKIS portrayal catalogues contains the specifications for the presentation of ALKIS primary database data (presentation output). It regulates the presentation of the AdV standard output. The ALKIS portrayal catalogues can be used as the foundation for the presentation of freely configurable outputs (variable in form and content)."

The Federal States' needs for freely arrangeable output exist. They are incorporated in many Federal State specific portrayal catalogues. Different solutions have been developed for meeting these needs and the following illustrate examples from two Federal States:

Baden-Württemberg is the only Federal State to accommodate Federal State specific parts with their own portrayal numbers in the AdV's ALKIS portrayal catalogue. An enhancement of the portrayal catalogue with Federal State specific parts is not being considered by the AdV as they are not responsible for Federal State specific specifications.

North Rhine-Westphalia have developed their own ALKIS portrayal catalogues whose content is published in parallel to the AdV.

The following regulation ensure the redundancy free nature of signature numbers in connection with Federal State specific enhancements to the ALKIS portrayal catalogue. This minimises the chance, in the future, of countrywide data users being confronted with identical portrayal numbers from several Federal States that could affect the contents of different presentations. The following regulations apply:

- The attribute "portrayal number" is mandatory for all Federal State specific presentation objects.
- Presentation rules that are accompanied by Federal State specific portrayal numbers are to be developed on a Federal State specific basis.
- Federal State specific portrayal numbers consist of the Federal State mnemonic (such as "BU" or "BKG" according to section 3.3.10 "Identifiers, Links", followed immediately by the four digit number of the portrayal number. They will be held in the AP_GPO.sgnaturnummer (portrayalnumber) (or the corresponding inherited object). Examples: RP4141, NW0311.

3.3.6 Map geometry objects

Map geometry objects are defined as those features whose geometric form and/or position have changed, for reasons of cartographic generalisation, during the derivation of a specific map scale. A map geometry object must contain the following independent information: The identifier, the map model data, e.g. DTK10, to which it belongs, the unidirectional relation *ist_abgeleitet_aus (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 in the derivation rules of the portrayal catalogue for the presentation.

3.3.7 Point Coverage Objects

A feature is defined as a point coverage object (Punktmengenobjekte (PMO) if it comprises a large number of geometric locations, each with identical attributes. In the AAA application context this is mainly the case for Digital Terrain Models, which comprise elevation values for defined points in a recitified grid. Additionally the capability is required for storing measured elevation values for arbitrary point locations. For this reason, as well as a class for gridded coverages (AD_GitterCoverage), a class for arbitrarily distributed points (AD_PunktCoverage) is modelled. The modelled classes implement the ISO 19123 Coverages classes with the following restriction: The sequence of the attribute values within the grid (CV_SequenceType) must be "linear".

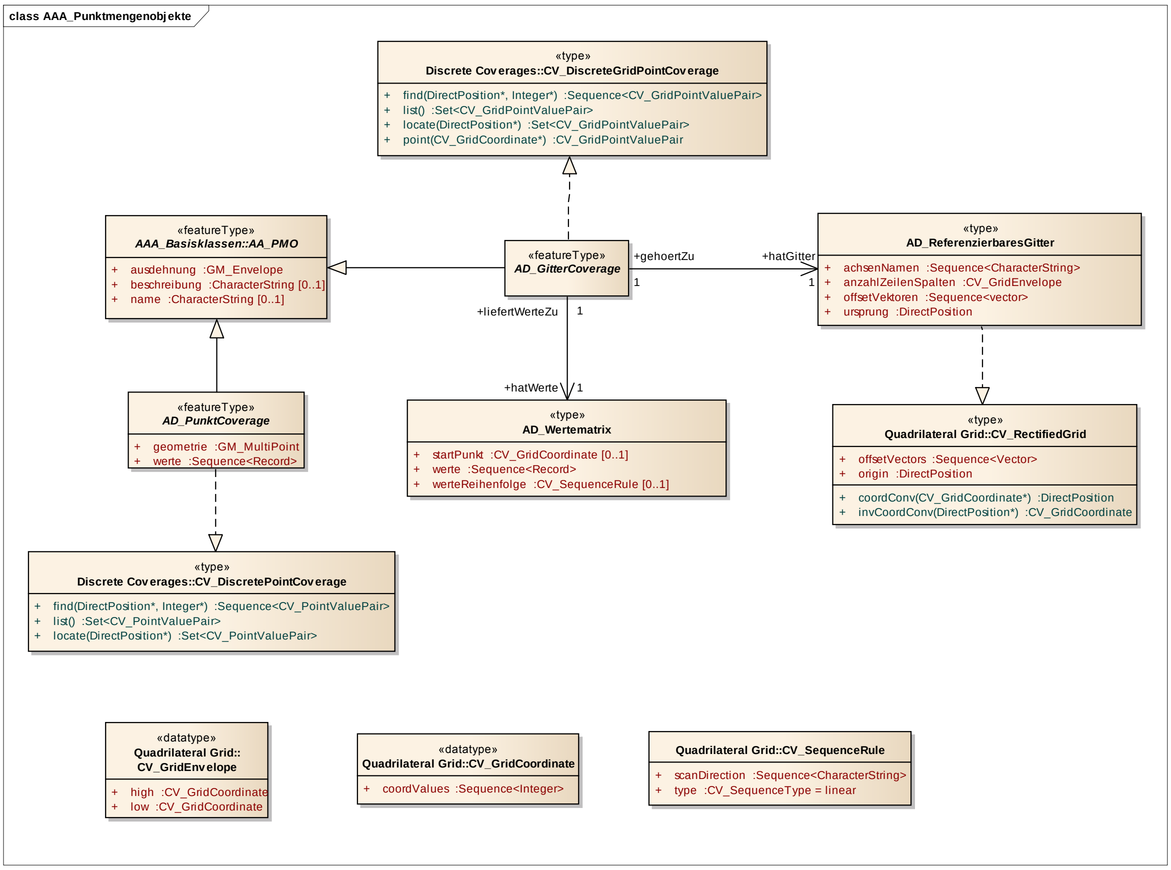

class AAA_Punktmengenobjekte
«type»
Discrete Coverages::CV_DiscreteGridPointCoverage
+ find(DirectPosition*, Integer*) :Sequence<CV_GridPointValuePair>
+ list() :Set<CV_GridPointValuePair>
+ locate(DirectPosition*) :Set<CV_GridPointValuePair>
+ point(CV_GridCoordinate*) :CV_GridPointValuePair
«featureType»
AAA_Basisklassen::AA_PMO
+ ausdehnung :GM_Envelope
+ beschreibung :CharacterString [0..1]
+ name :CharacterString [0..1]
«featureType»
AD_GitterCoverage
+gehoertZu
1
+hatGitter
1
«type»
AD_ReferenzierbaresGitter
+ achsenNamen :Sequence<CharacterString>
+ anzahlZeilenSpalten :CV_GridEnvelope
+ offsetVektoren :Sequence<vector>
+ ursprung :DirectPosition
+liefertWerteZu
1
+hatWerte
1
«featureType»
AD_PunktCoverage
+ geometrie :GM_MultiPoint
+ werte :Sequence<Record>
«type»
AD_Wertematrix
+ startPunkt :CV_GridCoordinate [0..1]
+ werte :Sequence<Record>
+ werteReihenfolge :CV_SequenceRule [0..1]
«type»
Quadrilateral Grid::CV_RectifiedGrid
+ offsetVectors :Sequence<Vector>
+ origin :DirectPosition
+ coordConv(CV_GridCoordinate*) :DirectPosition
+ invCoordConv(DirectPosition*) :CV_GridCoordinate
«type»
Discrete Coverages::CV_DiscretePointCoverage
+ find(DirectPosition*, Integer*) :Sequence<CV_PointValuePair>
+ list() :Set<CV_PointValuePair>
+ locate(DirectPosition*) :Set<CV_PointValuePair>
«datatype»
Quadrilateral Grid::
CV_GridEnvelope
+ high :CV_GridCoordinate
+ low :CV_GridCoordinate
«datatype»
Quadrilateral Grid::CV_GridCoordinate
+ coordValues :Sequence<Integer>
Quadrilateral Grid::CV_SequenceRule
+ scanDirection :Sequence<CharacterString>
+ type :CV_SequenceType = linear

Figure 18: Modelling of the Point Coverage Classes

3.3.8 Regulations for countrywide, redundancy free allocation of Federal State specific associations to technical data

In all AAA objects the attribute "pointsToExternal" (zeigtAufExternes) can contain an association to technical data. The model allows two forms, either a URN or a URL. In the model the following is noted against the attribute AA_association_to_technical data.type (AA_Fachdatenverbindung.art):

Documentation
This attribute type defines the namespace for specifying the type of the association to technical data.
Use URNs when dealing with a non-generally solvable namespace. When URLs are used the resource referenced must return a description of the association to technical data.
URLs must use the HTTP protocol.

Associations to technical data that use the URL variant are not a problem due to the uniqueness of URNs.

The following regulations ensure the freedom from redundancy of associations to technical data freedom in State specific expansion with the URN variant. This minimises the chance, in the future, of countrywide data users being confronted with identical numbers of associations to technical data from several Federal States that actually have different factual contents.

Following the normal URN logic in the GeoInfoDoc

- urn:adv:uom for units of measure
- urn:adv:crs for coordinate reference systems
- urn:adv:oid for object identifiers

the following URNs are to be used for associations to technical data:

- urn:<State abbr.>:fdv:<four character code>

where <Stat abbr> should be formed according to section 3.3.10 Identifiers, Links – but in lowercase – and <four character code>is always four characters from digits 0-9 (with leading zeros).

Examples: urn:rp:fdv:4711 bzw. urn:by:fdv:0203.

In this way the uniqueness of URN association to technical data variants is ensured. Development of the codes and their referenced content are the responsibility of the respective States.

3.3.9 Identifiers, Links

Identifiers act on behalf of the object that they represent. For this reason they are also called **object identifiers**, or **OID** for short. 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 countrywide and theme-wide uniqueness.
- Its appearance indicates that an object has been generated.
- It remains unchanged during the lifetime of an object.
- Its disappearance 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 or may not be changed cannot therefore be answered from an IT point of view; technical criteria must be stated:

- when is an object created,
- what changes can it withstand without losing its identity and

- when does it disappear.

So object identifiers of objects that have been created by other departments within the scope of their responsibilities are also adopted in unmodified form when transferred to the AAA data storage. For all thematic 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 provisional identifiers.	The characters start from the left with the abbreviations of the German states standardised in ISO 3166-2 "Country Subdivision Code" (ISO, 15th December 1998). For Federal Agencies, the abbreviation "BU" is used 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 Cartograpy and Geodesy (Bundesamt für Kartographie und Geodäsie) "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" (final identifier)
"DE_000000000001" (provisional identifier)

When implementing a geodata infrastructure as defined by the Open Geospatial Consortium (OGC) and using its interface definitions, all states and agencies involved 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 provided as a countrywide solution; the system of assignment and distribution can still be defined state-specifically.

In order to construct relations between features in data exchange, identifiers are also used as references to features.

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 object version addressed.

An important precondition for the joint management of databases of various origins is that the integration situation is represented by means of references between the data of the surveying and mapping authority and the technical data (**link**). This linking can take place unilaterally in the spatially-referenced 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 defined and managed. These may also consist of the aforementioned identifiers and / or technical codes of the respective databases.

3.3.10 Model Type

The feature class AA_Object (AA_Objekt) includes the attribute *model type,*which represents an assignment of one or more model types. For a 3D technical model, corresponding model types have been added. The extendable code list “AA_OtherModelType” (AA_WeitereModellart) allows the capture of any desired model types, e.g. Federal State specific ones. The entry and maintenance of such value types and identifiers is regulated within the scope of the AdV Registry for Code Lists.

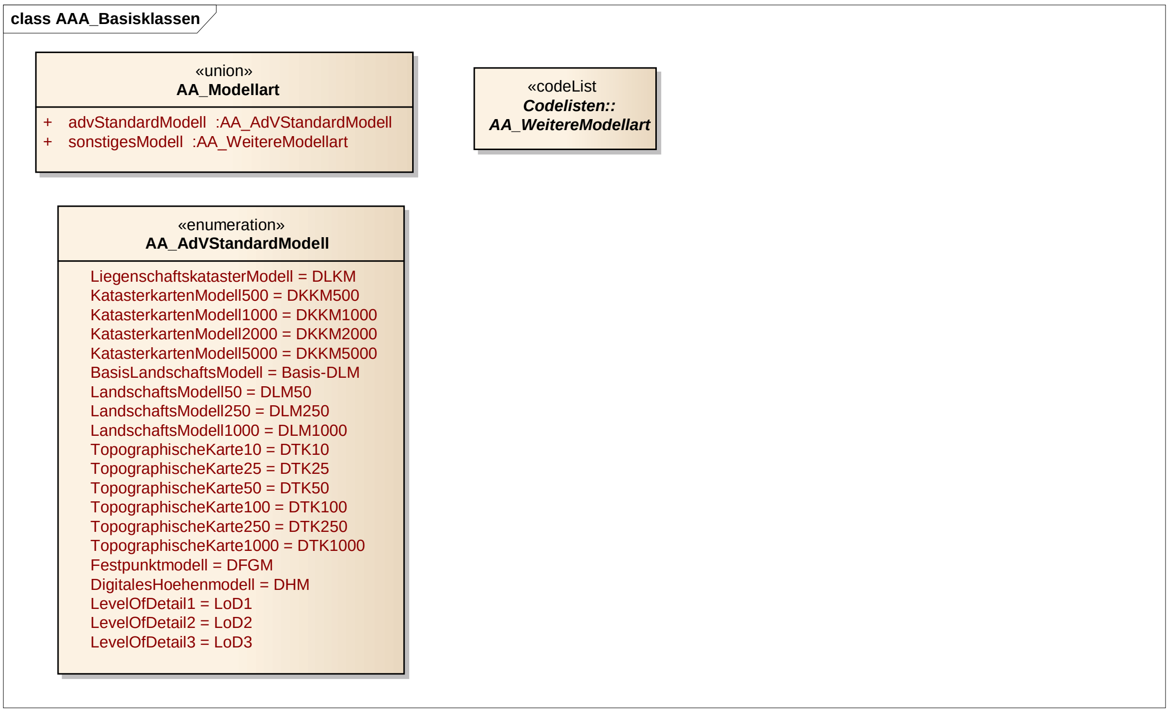

class AAA_Basisklassen
«union»
AA_Modellart
+ advStandardModell :AA_AdVStandardModell
+ sonstigesModell :AA_WeitereModellart
«codeList
Codelisten::
AA_WeitereModellart
«enumeration»
AA_AdVStandardModell
LiegenschaftskatasterModell = DLKM
KatasterkartenModell500 = DKKM500
KatasterkartenModell1000 = DKKM1000
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
DigitalesHoehenmodell = DHM
LevelOfDetail1 = LoD1
LevelOfDetail2 = LoD2
LevelOfDetail3 = LoD3

Figure 19: Model types in the Basic Schema

On the other hand, the enumeration AA_AdVStandardModell is not expandable and includes the approved model types for the AFIS, ALKIS and ATKIS application schemas. Using this model type, all elements of the data model (e.g. feature types, attribute types, etc.) can be assigned to multiple models. Thus, despite the unified, integrated modelling, different technical views of the object can be represented and output as technically-specific features catalogues. This model type has no effect on the NAS, however. The NAS content is defined via the whole AAA application schema, whereby, at the interface level, no individual, technical views can be represented. That means there is neither an AKTIS-NAS nor an ALKIS-NAS but just **the** NAS.

3.3.11 LoD Definition

The Level of Detail describes the detailing level of the 3D geometry of a spatially-referenced elementary object. This is usually defined by means of the capture or derivation method for the 3D geometry. Only the Levels of Detail 1 to 3 should be used for ALKIS 3D. The inverse relation role "detailed" (detailliert) refers to the associated spatially-referenced elementary object with a 3D geometry in a lower detailing level. The inverse relation role "generalised" (generalisiert) refers to the associated spatially-referenced elementary object with a 3D geometry in a higher detailing level.

The 3D extension supports various Levels of Detail (LoD). Lodes are needed to allocate a certain detailing level to buildings and other 3D objects. They also help visualisation and data analysis.

The following documents are drawn on for the definition of the individual Lodes.

- 07-062_Candidate_OpenGIS_CityGML_Implementation_Specification.pdf
- 3D City Models, An orientation to the AG City Models of the workgroup of the Community Surveying and Real Estate Association of NRW (DE: Eine Orientierungshilfe der AG Stadtmodelle des AK KommunalesVermessungs- und Liegenschaftswesens des Städtetages NRW)

LoD1, the lowest level, is the block model where buildings are portrayed as simple blocks with flat roofs. LoD2 portrays the different types of roof, vegetation can be portrayed. LoD3 has the highest detailing level. There details of wall and roof structures, vegetation and road furnishings are represented. As well as visual criteria, the minimum geometric requirements are defined by the Lodes (see table):

- the absolute location and elevation accuracy
- the ground area of the object to be portrayed.

	LoD1	LoD2	LoD3
Absolute location / elevation accuracy (better than ...)	5/5m	2/1m	0.5/0.5m
Portrayal	Object blocks as generalised shapes; > 6*6m/3m	Objects as generalised shapes; > 4*4m/2m	Objects in real shapes; > 2*2m/1m
Roof type	flat roof	roof type and orientation	real shape
foreign objects (road furniture)	important objects	prototypes	real shape

Table: LoD 1 - 3 with geometric accuracies

A complete textured LoD will be available shortly. For this reason texturings do not represent a criteria for the classification of a specific LoD and are allowed in all Lodes.

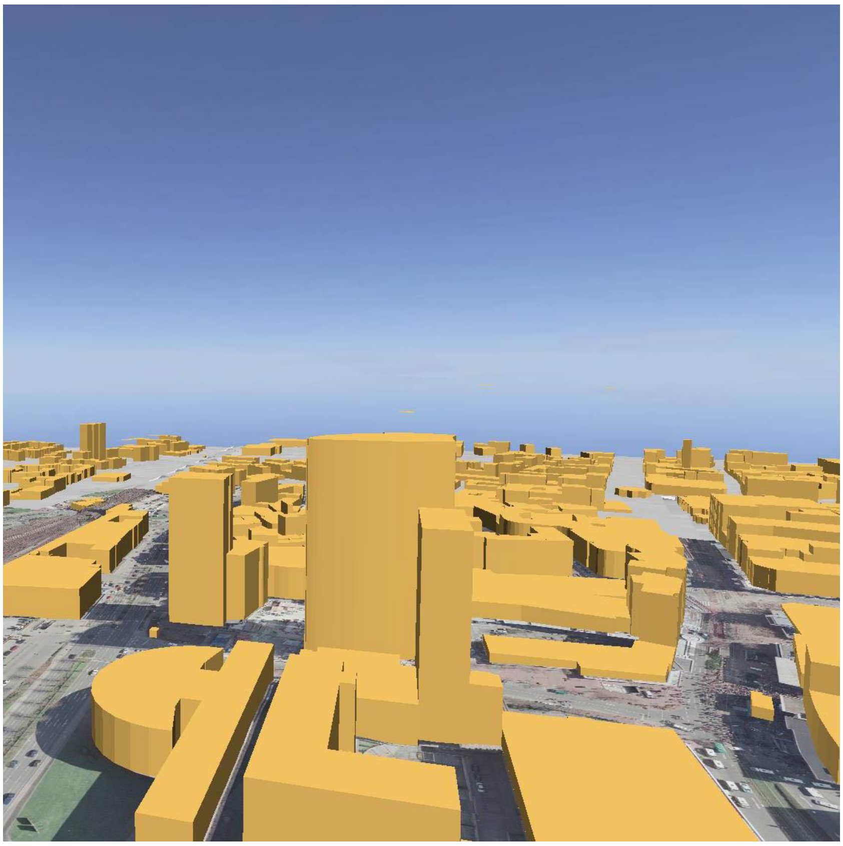


Figure 20: LoD 1

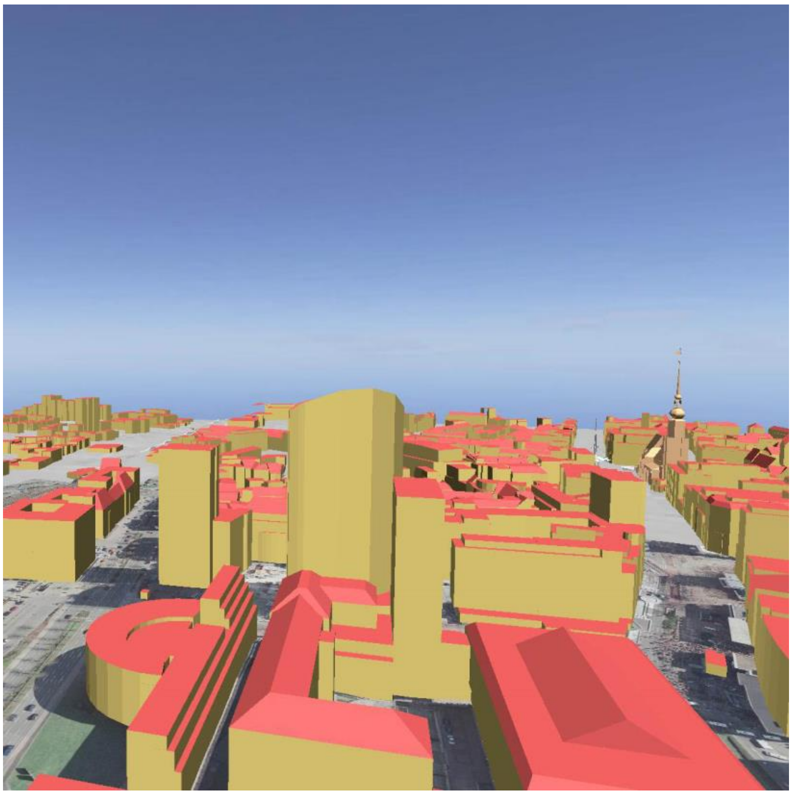


Figure 21: LoD 2

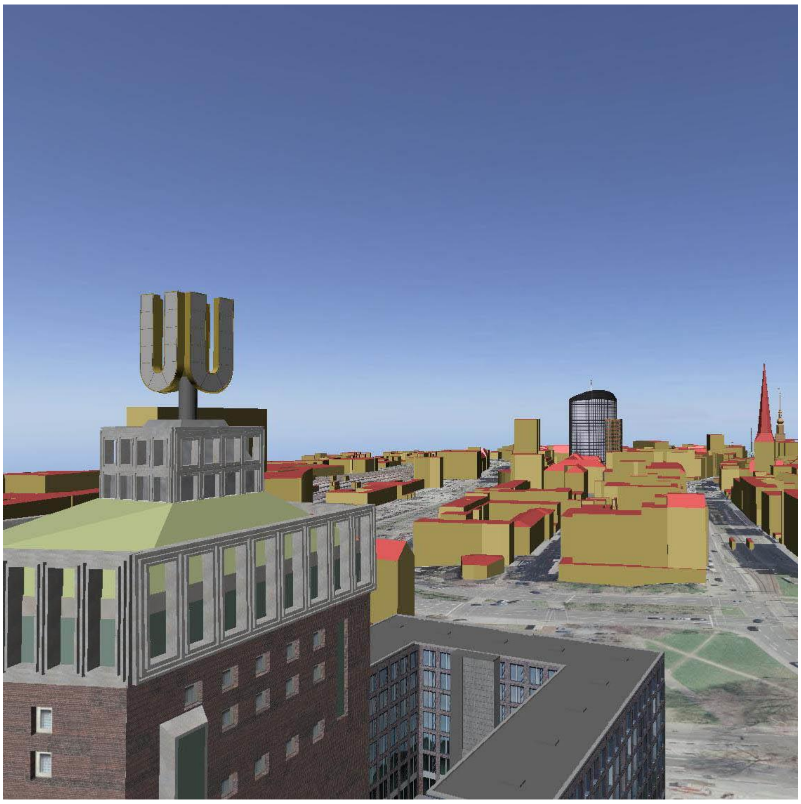


Figure 22: LoD 3

3.3.12 Use of GeometryLibraries

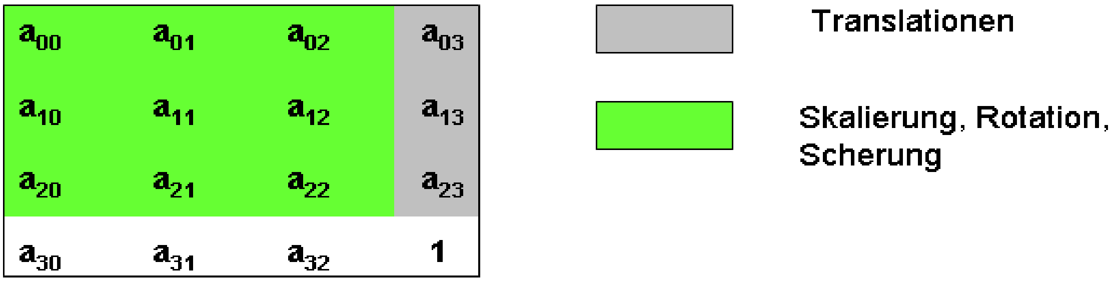
Geometry libraries are used to extend the capabilities of geometric imaging in the 3D area. Geometry libraries can be used to integrate prototypes. Examples in the 3D area include trees, traffic lights, lamps, etc. Each geometric prototype exists only once in the Cartesian coordinate system and can be referenced multiple times using the URI of a special presentation object. The prototypical geometry can be in a local file, a remote file or delivered by a web service. The type of geometry referenced is described via its attributes. The visualisation software must guarantee the correct presentation of this Mime type. A 3D transformation matrix is used to specify the positioning of the prototypical geometry in the data inventory It includes the 16 parameters for rotation, translation and scaling to the local geometry. The geometry is a *GM_MultiPoint* enabling objects with the same attributes to be set multiple times in the data inventory.

Use of geometry libraries has some advantages over the explicit referencing of objects using absolute coordinates:

- more efficient use of memory than with explicit geometry,
- complex scenes can be processed, and

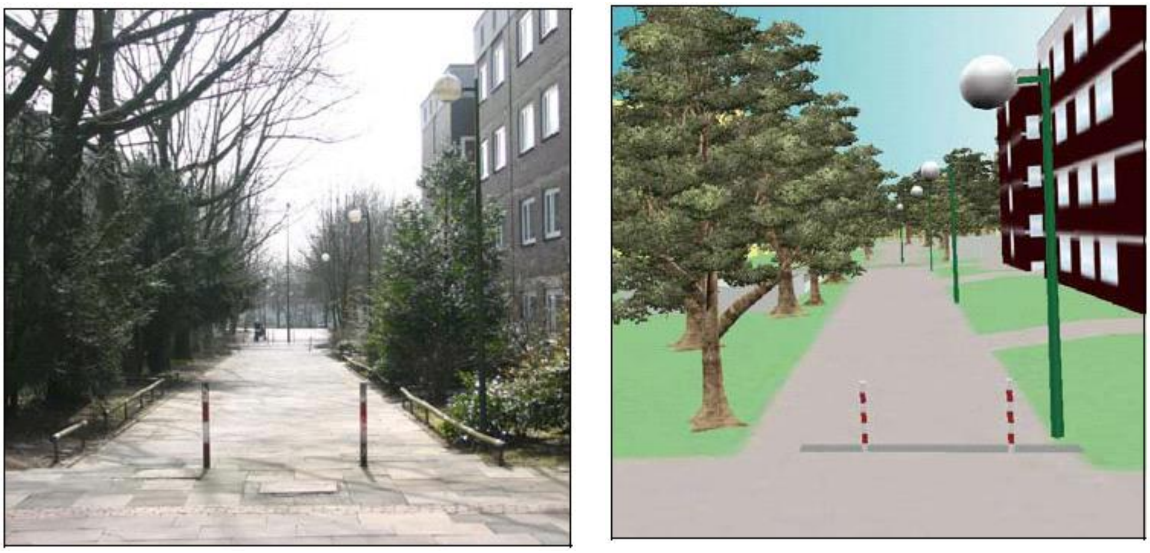
- flexible alteration of the characteristics of referenced objects (exchange of library objects).

3D-Transformations Matrix



$$\begin{array}{cccc} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & 1 \end{array}$$

The data type *AP_TransformationsMatrix_3D* represents a general transformation matrix as a vector. The principle is based on the scene graph concept that is in common use in computer graphics. The matrix contains all parameters required for the transformation from homogenous coordinates from any right-angled coordinate system of the prototype to the coordinate system of the 3D model.



Real-world Scene

3D picture using geometry libraries (boundary posts, lamps)

Figure 23: Example of the use of geometry libraries.

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 federal states. One essential application for the version concept is the procedure for user-related updating of the primary database (NBA).

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

- In the application schema, no distinction is made 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 (versions) 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 feature carries an identifier, attribute and relation, as well as a lifetime interval (creation and expiry date). Creation and expiry dates include the date and time with an accuracy of one second. Entering an object into the primary database data generates the first version of the object and registers it in a container for feature versions. If a non-object forming property changes due to an updating, a new version of the object is generated. The historicized 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 an update, this results, from a technical point of view, in the expiry of an object. The object is historicized 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 feature view always remains in place.

Definition of Object Forming Properties in the AAA Model in UML

Attribute and relation statements are stored in the AAA model in UML, each with a definition of the affects of updates. These are held in the tab "Tagged Values" in the "AAA objectforming" (objektbildend) field of the Attribute/Relation and have a value of True or False. All entries in the officially published AAA model are managed by the AdV. The following explanations describe the definitions:

Definition in the UML Model (Rider AAA / *object forming* to the respective Attribute / Relation)	**Status of the definition in rider AAA / object forming**
True	unalterable AdV definition
False	predefined framework from AdV, i.e it can be set on a state specific basis to True or False*

(*this opportunity to accommodate state specific definitions in the AAA model is a unique exception).

Definition in the UML Model (Rider AAA / *object forming* to the respective Attribute / Relation)	**Effect of the definition from column 1 when changing affected Attributes / Relations in a creation component**
True	New object is created (Delete + Insert)
False	New object version is created (Replace)

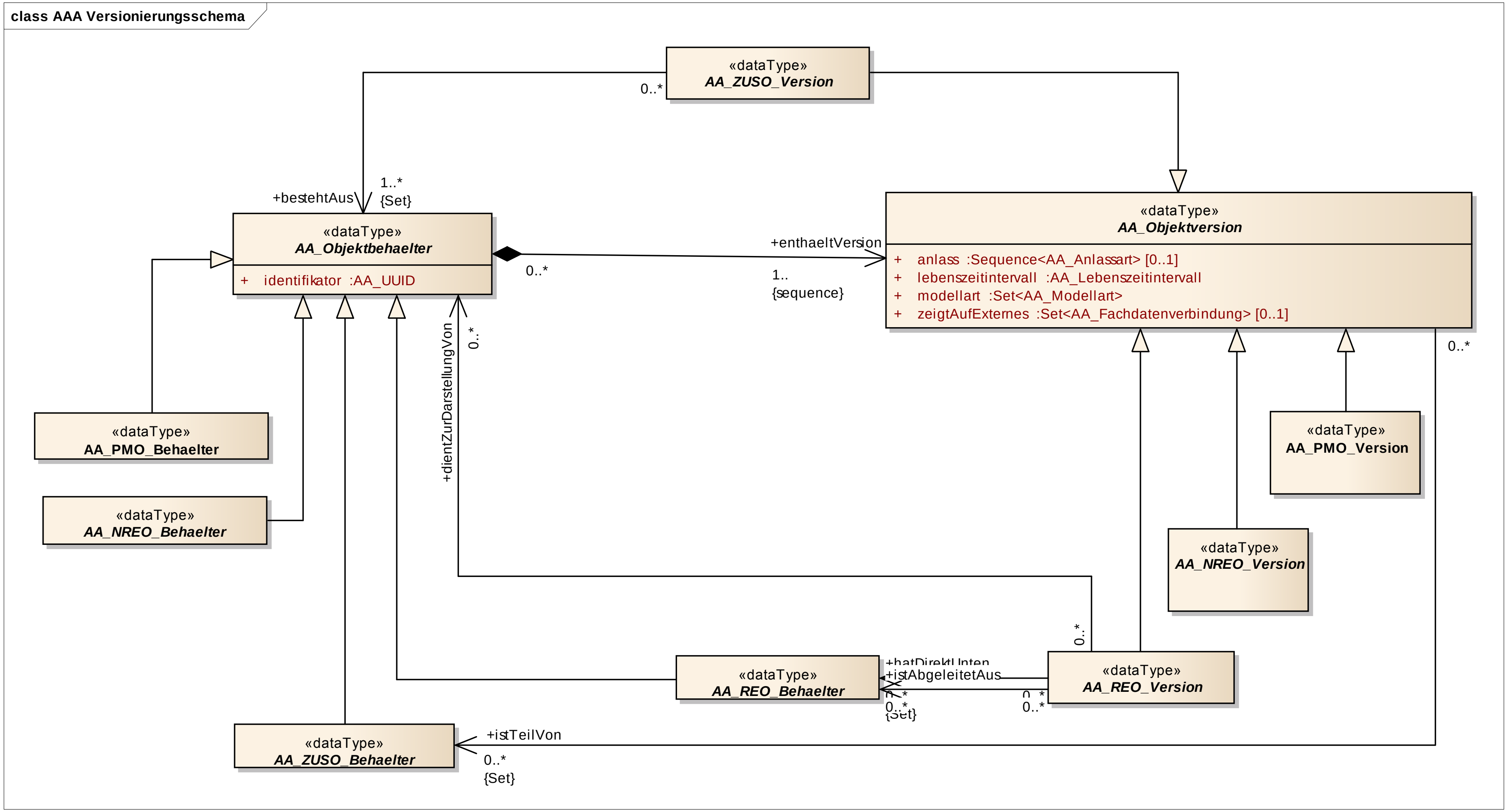


Figure 24: Versioning schema

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:

	Identifikator	Time interval		Name	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 or version is in the future. At time t_2, Mrs Huber changes her name to Meier, i.e. due to the change to the attribute *Name* a new version of the feature *Person* is created from object "DEBU5t44dFzb70Lg".

	Identifikator	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 point in time when Version 1 expires is identical to the creation date when Version 2 of the object is created. At time t_3, Mrs Meier sells her only plot of land. Because she has no other role within ALKIS, the object expires from a technical viewpoint

	Identifikator	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 historicized, 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.

That is explained using the example described above: Mrs Hilde Huber, address Ottostrasse 17 in 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 is 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, as 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 an 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.

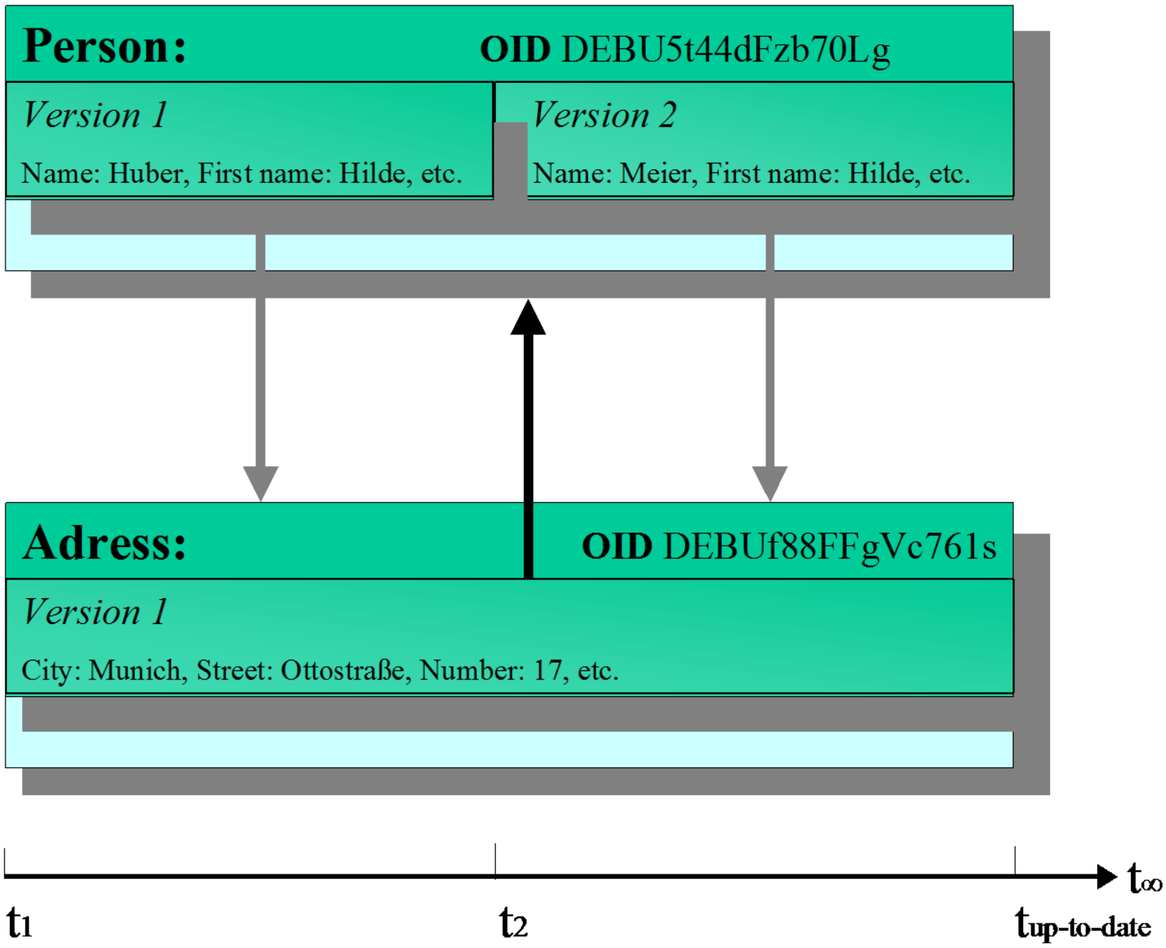

Person:
OID DEBU5t44dFzb70Lg
Version 1
Name: Huber, First name: Hilde, etc.
Version 2
Name: Meier, First name: Hilde, etc.
Adress:
OID DEBUf88FFgVc761s
Version 1
City: Munich, Street: Ottostraße, Number: 17, etc.
t∞
t1
t2
tup-to-date

Figure 25: Example of versioning following change of attributes

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

Changes to relations

Changes to relations or attributes both result in versioning of objects. 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 29. At time t₃, Mrs Hilde Huber moves from 17 Ottostraße, Munich to Platanenallee 34a, Berlin The *Address* object with OID "DEBUf88FFgVc761s", to which the *has_address* relation points from the *Person* object, is exchanged (new OID "DEBUk41233THjbkO"). Thus, the relation associated with the *Person* object changes and hence the *Person* object must be versioned.

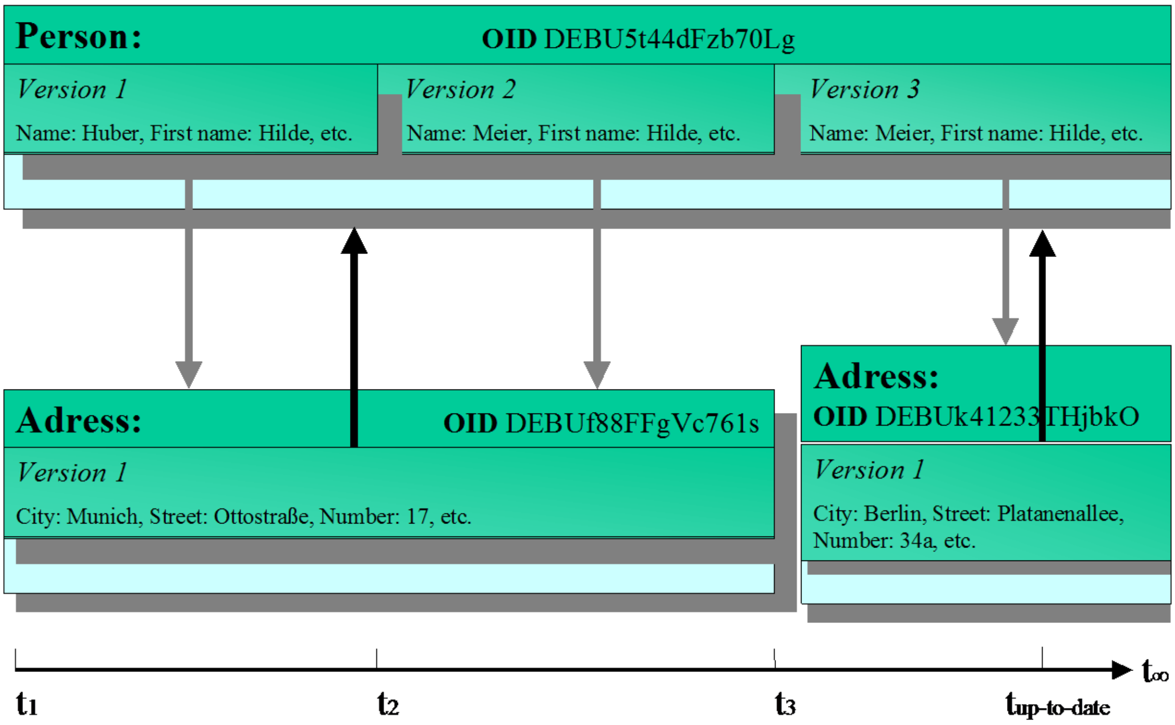

Person:
OID DEBU5t44dFzb70Lg
Version 1
Name: Huber, First name: Hilde, etc.
Version 2
Name: Meier, First name: Hilde, etc.
Version 3
Name: Meier, First name: Hilde, etc.
Adress:
OID DEBUf88FFgVc761s
Version 1
City: Munich, Street: Ottostraße, Number: 17, etc.
Adress:
OID DEBUk41233THjbkO
Version 1
City: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

Figure 26: Example of versioning following change of relations

Illustrated in tabular form:

	Identifikator	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 Data and Metadata

The common AFIS-ALKIS-ATKIS data model records and manages quality data 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 sub-divided into non-quantifiable summary information (purpose, application and history) and quantifiable information (data quality elements *completeness*, *logical consistency* as well as *geometric*, *content and temporal accuracy*).

The quality information is detailed as metadata in accordance with ISO 19115 and additionally, for quantitative, aggregated quality data if required, in the form of detailed quality evaluation protocols in accordance with ISO 19114, which is, however, not part of the GeoInfoDoc and is thus not further specified here.

Example of quality information about a point location with the following properties:

- data collection 'determined from cadastral survey (1000)'
(Datenerhebung 'Aus Katastervermessung ermittelt (1000)')
- collection date '01.04.1990'° (Erhebungsdatum '01.04.1990')
- collection office Land Registry X (Erhebungsstelle Katasteramt X)
- calculation date '01.01.94'° (Berechnungsdatum '01.01.1994')
- no details about calculating office (keine Angabe zur berechnenden Stelle)
- accuracy 2.2 cm° (Genauigkeitswert 2,2 cm)
- accuracy level 2000° (Genauigkeitsstufe 2000)
- trustworthiness 1200° (Vertrauenswürdigkeit 1200)

Guidelines:

- According to GeoInfoDoc 4.4.2 (last bullet) and 4.4.4 first sentence and ISO/TS 19139 8.5.4 referenced there, the specific element from the AAA thematic schema which substitutes gco:CharacterString is to be used for enumerations.
- If a source for a process step is given, this is embedded in the LI_ProcessStep, in order to enable an assignment.
- Provided a location for creation or calculation is given, the role "processor" is to be declared.
- In the role information a code list pointer is required that, according to ISO/TS 19139 8.5.5, must be a URL. In the example a URL of a Code-List_Dictionary in OGC-Schema repository is given. Alternatively, this can also be - as with schema pointers - another valid pointer to a Code-List-Dictionary.
- The name of location responsible is given in clear text.
- Only units given in GeoInfoDoc 8.2.2 can be used for units of accuracy. The unit "m" is used according to GeoInfoDoc 8.2.3 "urn:adv:uom:m".

- According to the example in ISO/TS 19139 7.4.1.4.d), with gco:Record, the data type in xsp:type should be specified. In the case of coordinate accuracy this should be "double" from the XML Schema.

Example:

```
<AX_DQPunktort
xmlns="http://www.adv-online.de/namespaces/adv/gid/6.0"
xmlns:gmd="http://www.isotc211.org/2005/gmd"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:gco="http://www.isotc211.org/2005/gco"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:schemaLocation="http://www.adv-online.de/namespaces/adv/gid/6.0
aaa.xsd">
  <herkunft>
   <gmd:LI_Lineage>
    <gmd:processStep>
     <gmd:LI_ProcessStep>
      <gmd:description>
       <AX_LI_ProcessStep_Punktort_Description>Erhebung</
AX_LI_ProcessStep_Punktort_Description>
      </gmd:description>
      <gmd:dateTime>
       <gco:DateTime>1990-04-01T00:00:00Z</gco:DateTime>
      </gmd:dateTime>
      <gmd:source>

<AX_Datenerhebung_Punktort>1000</AX_Datenerhebung_Punktort>
             </gmd:source>
      <gmd:processor>
       <gmd:CI_ReponsibleParty>
        <gmd:organisationName>
         <gco:CharacterString>Katasteramt  X</gco:CharacterString>
        </gmd:organisationName>
       <gmd:role>
        <gmd:CI_RoleCode
codeList="http://schemas.opengis.net/iso/19139/20070417/resources/Codelis
t/gmxCodelists.xml#CI_RoleCode"
codeListValue="processor">processor</CI_RoleCode>
        </gmd:role>
       </gmd:CI_ReponsibleParty>
      </gmd:processor>
     </gmd:LI_ProcessStep>
    </gmd:processStep>
    <gmd:processStep>
     <gmd:LI_ProcessStep>
      <gmd:description>

<AX_LI_ProcessStep_Punktort_Description>Berechnung</AX_LI_ProcessStep_Pun
ktort_Description>
      </gmd:description>
      <gmd:dateTime>
       <gco:DateTime>1994-01-01T00:00:00Z</gco:DateTime>
      </gmd:dateTime>
     </gmd:LI_ProcessStep>
    </gmd:processStep>
   </gmd:LI_Lineage>
  </herkunft>
  <genauigkeitswert>
   <gmd:DQ_RelativeInternalPositionalAccuracy>
    <gmd:result>
     <gmd:DQ_QuantitativeResult>
     <gmd:valueUnit  xlink:href="urn:adv:uom:m"/>
      <gmd:value>
       <gco:Record  xsi:type="xsd:double">0.022</gco:Record>
```

```
        </gmd:value>
      </gmd:DQ_QuantitativeResult>
    </gmd:result>
   </gmd:DQ_RelativeInternalPositionalAccuracy>
  </genauigkeitswert>
  <genauigkeitsstufe>2000</genauigkeitsstufe>
  <vertrauenswuerdigkeit>1200</vertrauenswuerdigkeit>
</AX_DQPunktort>
```

Metadata is "data about data" and describes geodata in terms of user-relevant aspects for evaluating the suitability of the data and access to that data. ISO differentiates between about 400 optional, mandatory and conditionally mandatory metadata elements, subdivided into semantic entities and also into the following sections:

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

In accordance with ISO, quality data and metadata can be specified for a data inventory (collection of logically associated objects), for reporting groups (subsets of a data inventory) or for individual objects.

The common AFIS-ALKIS-ATKIS metadata catalogue is described in more detail below.

3.6 Feature Catalogues

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; in the AAA data model, however, this has so far rarely been made use of. The guidelines on the structure of a feature catalogue are to be seen as logically separate from the description of those feature types in a data model. As a result, the schema “AAA feature catalogue is modelled independently outside the AAA application schema. The same applies to the guidelines on the description of the signatures and presentation rules. The common application schema expands these structures in the *AAA Catalogue)* package around the areas that are additionally required for the AFIS, ALKIS and ATKIS applications.

To achieve a simpler implementation, catalogues are versioned and exchanged as complete items only.

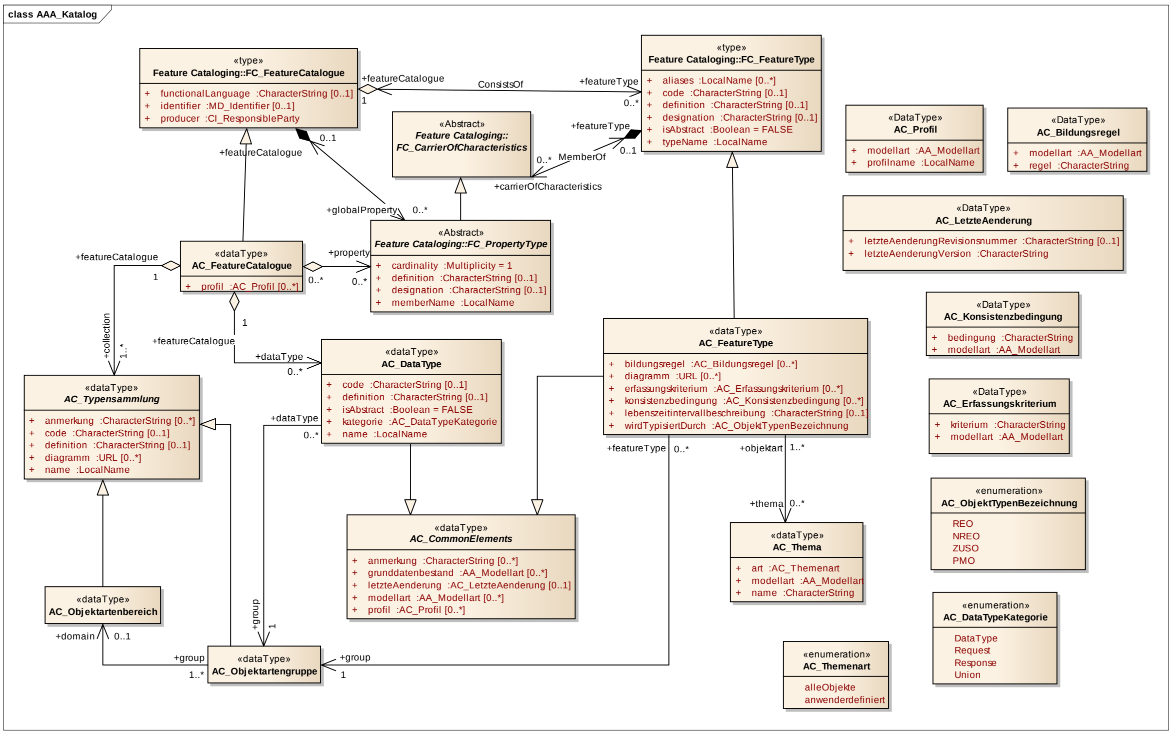

class AAA_Katalog
«type»
Feature Cataloging::FC_FeatureCatalogue
+ functionalLanguage :CharacterString [0..1]
+ identifier :MD_Identifier [0..1]
+ producer :CI_ResponsibleParty
+featureCatalogue
1
ConsistsOf
+featureType
0..*
«type»
Feature Cataloging::FC_FeatureType
+ aliases :LocalName [0..*]
+ code :CharacterString [0..1]
+ definition :CharacterString [0..1]
+ designation :CharacterString [0..1]
+ isAbstract :Boolean = FALSE
+ typeName :LocalName
«Abstract»
Feature Cataloging::
FC_CarrierOfCharacteristics
+featureType
0..1
MemberOf
0..*
+carrierOfCharacteristics
0..1
+featureCatalogue
+globalProperty
0..*
«DataType»
AC_Profil
+ modellart :AA_Modellart
+ profilname :LocalName
«DataType»
AC_Bildungsregel
+ modellart :AA_Modellart
+ regel :CharacterString
«DataType»
AC_LetzteAenderung
+ letzteAenderungRevisionsnummer :CharacterString [0..1]
+ letzteAenderungVersion :CharacterString
+featureCatalogue
1
«dataType»
AC_FeatureCatalogue
+ profil :AC_Profil [0..*]
+property
0..*
0..*
«Abstract»
Feature Cataloging::FC_PropertyType
+ cardinality :Multiplicity = 1
+ definition :CharacterString [0..1]
+ designation :CharacterString [0..1]
+ memberName :LocalName
«DataType»
AC_Konsistenzbedingung
+ bedingung :CharacterString
+ modellart :AA_Modellart
1
+featureCatalogue
+collection
1..*
+dataType
0..*
«dataType»
AC_DataType
+ code :CharacterString [0..1]
+ definition :CharacterString [0..1]
+ isAbstract :Boolean = FALSE
+ kategorie :AC_DataTypeKategorie
+ name :LocalName
«dataType»
AC_FeatureType
+ bildungsregel :AC_Bildungsregel [0..*]
+ diagramm :URL [0..*]
+ erfassungskriterium :AC_Erfassungskriterium [0..*]
+ konsistenzbedingung :AC_Konsistenzbedingung [0..*]
+ lebenszeitintervallbeschreibung :CharacterString [0..1]
+ wirdTypisiertDurch :AC_ObjektTypenBezeichnung
«DataType»
AC_Erfassungskriterium
+ kriterium :CharacterString
+ modellart :AA_Modellart
«dataType»
AC_Typensammlung
+ anmerkung :CharacterString [0..*]
+ code :CharacterString [0..1]
+ definition :CharacterString [0..1]
+ diagramm :URL [0..*]
+ name :LocalName
+dataType
0..*
+featureType
0..*
+objektart
1..*
+thema
0..*
«enumeration»
AC_ObjektTypenBezeichnung
REO
NREO
ZUSO
PMO
«dataType»
AC_Thema
+ art :AC_Themenart
+ modellart :AA_Modellart
+ name :CharacterString
«dataType»
AC_CommonElements
+ anmerkung :CharacterString [0..*]
+ grunddatenbestand :AA_Modellart [0..*]
+ letzteAenderung :AC_LetzteAenderung [0..1]
+ modellart :AA_Modellart [0..*]
+ profil :AC_Profil [0..*]
«dataType»
AC_Objektartenbereich
+domain
0..1
+group
1
«enumeration»
AC_DataTypeKategorie
DataType
Request
Response
Union
+group
1..*
«dataType»
AC_Objektartengruppe
+group
1
«enumeration»
AC_Themenart
alleObjekte
anwenderdefiniert

Figure 27: Expansion of standardised structure of the feature catalogues

The feature catalogue is derived, with the help of a script, directly from the UML data model and published in editable or readable formats in PDF and HTML. For each of the three information systems, AFIS, ALKIS and ATKIS, there is a document that explains about the technical stipulations in the feature catalogue and the data model as well as the theme specific portrayal catalogue. Portrayal catalogues, like feature catalogues, will be described in a standardised style in future. In addition, AFIS and ALKIS have output catalogues that can be derived, standards conformant, from the UML data model and the contents of the respective output described. The functionalities for creating these outputs are described with filter encoding (see below).

3.7 Portrayal Catalogues

The guidelines on the structure of a portrayal catalogue (SK) are of a superordinate nature and are therefore to be seen as logically separate from the description of technical feature types in a data model. As a result, the schema "AAA-Signaturenkatalog" (AAA portrayal catalogue) is modelled independently outside the AAA application schema.

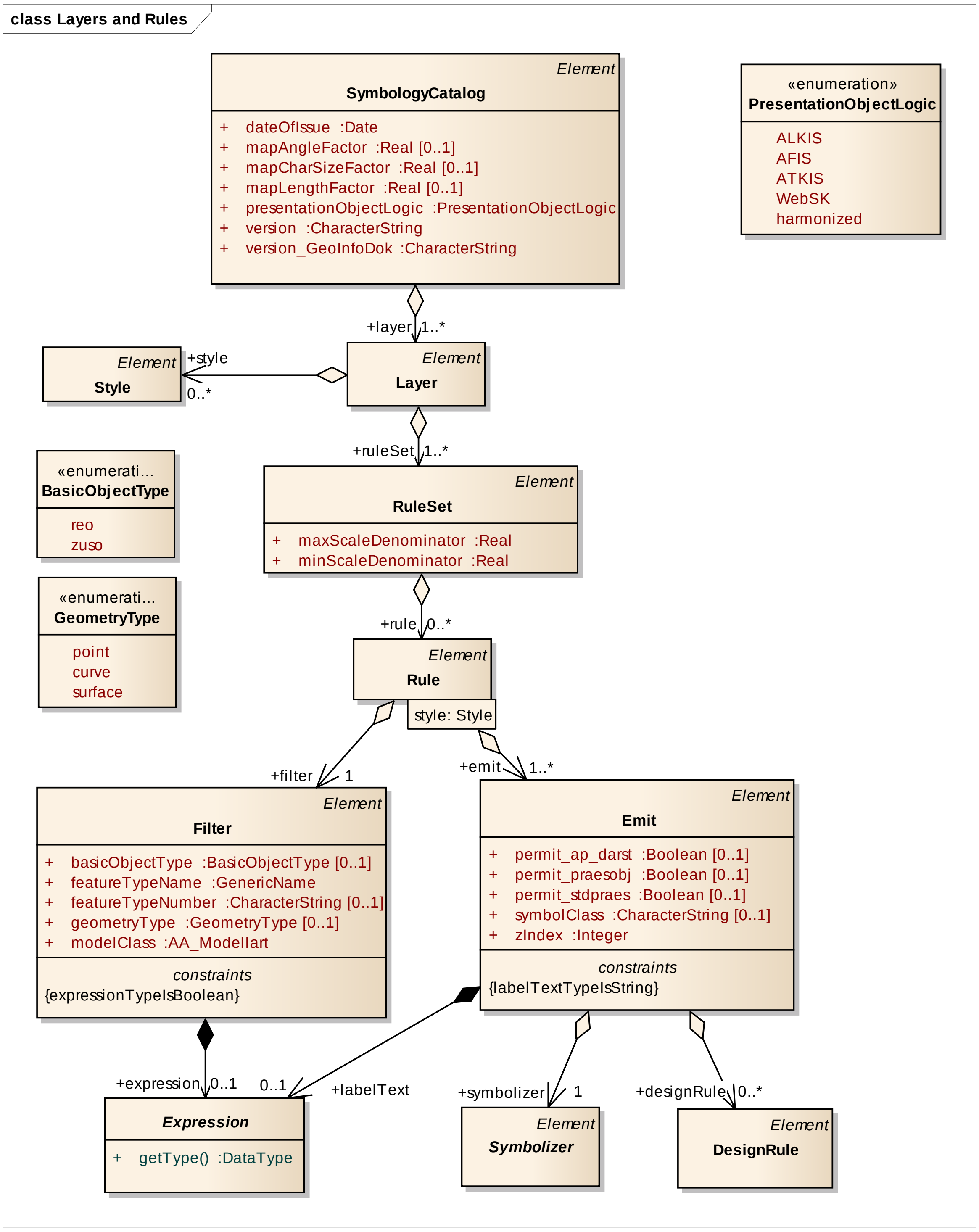

class Layers and Rules
Element
SymbologyCatalog
+ dateOfIssue :Date
+ mapAngleFactor :Real [0..1]
+ mapCharSizeFactor :Real [0..1]
+ mapLengthFactor :Real [0..1]
+ presentationObjectLogic :PresentationObjectLogic
+ version :CharacterString
+ version_GeoInfoDok :CharacterString
«enumeration»
PresentationObjectLogic
ALKIS
AFIS
ATKIS
WebSK
harmonized
+layer
1..*
Element
Style
+style
0..*
Element
Layer
+ruleSet
1..*
«enumerati...
BasicObjectType
reo
zuso
Element
RuleSet
+ maxScaleDenominator :Real
+ minScaleDenominator :Real
«enumerati...
GeometryType
point
curve
surface
+rule
0..*
Element
Rule
style: Style
+filter
1
+emit
1..*
Element
Filter
+ basicObjectType :BasicObjectType [0..1]
+ featureTypeName :GenericName
+ featureTypeNumber :CharacterString [0..1]
+ geometryType :GeometryType [0..1]
+ modelClass :AA_Modellart
constraints
{expressionTypeIsBoolean}
Element
Emit
+ permit_ap_darst :Boolean [0..1]
+ permit_praesobj :Boolean [0..1]
+ permit_stdpraes :Boolean [0..1]
+ symbolClass :CharacterString [0..1]
+ zIndex :Integer
constraints
{labelTextTypeIsString}
+expression
0..1
0..1
+labelText
+symbolizer
1
+designRule
0..*
Expression
+ getType() :DataType
Element
Symbolizer
Element
DesignRule

Figure 28: Schema of SK object model

The core of the modelling is an SK definition language, *Map Definition Language* ("MDL", originally called "SDL"), which can display all the SK definitions in a readable but nonetheless automatically processable form. A central MDL repository stores the status of the SK in MDL and thus defines the reference version of the SK.

An MDL compiler automatically derives SK documents in HTML and PDF from the SK defined in MDL. An additional output of the MDL compiler is a representation of the SK in "SK XML", an XML language that essentially uses the same content as MDL. Representation in SK XML is intended to make it possible to generate external representations of the SK from this in a simple way in order to support commercially available GIS software. The implementations from SK XML are to be conducted by external parties.

Since MDL and SK XML are, by nature, closely related structurally, it seems reasonable to develop both of them on the basis of a common object model defined in UML.

3.8 Processes, Operations and Activities (Methods)

3.8.1 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 results in one or more processes. There are collection, qualification, management, use and transfer processes.

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

Project control in the AAA basic schema controls the operations and activities that form processes and from which complete business processes can be described. They represent simply an optional guideline, the contents of which can be used to underpin the state-specific business processes. Figure 34 illustrates 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.

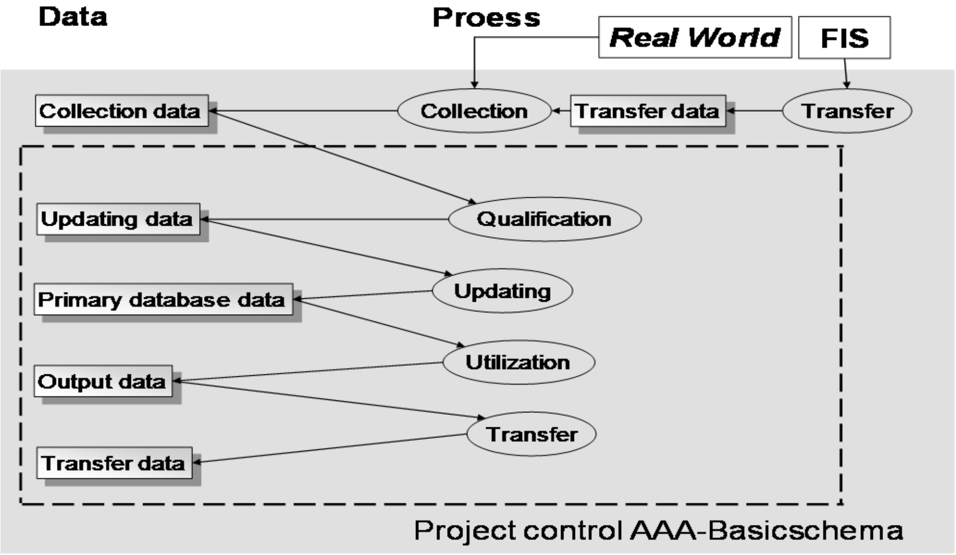


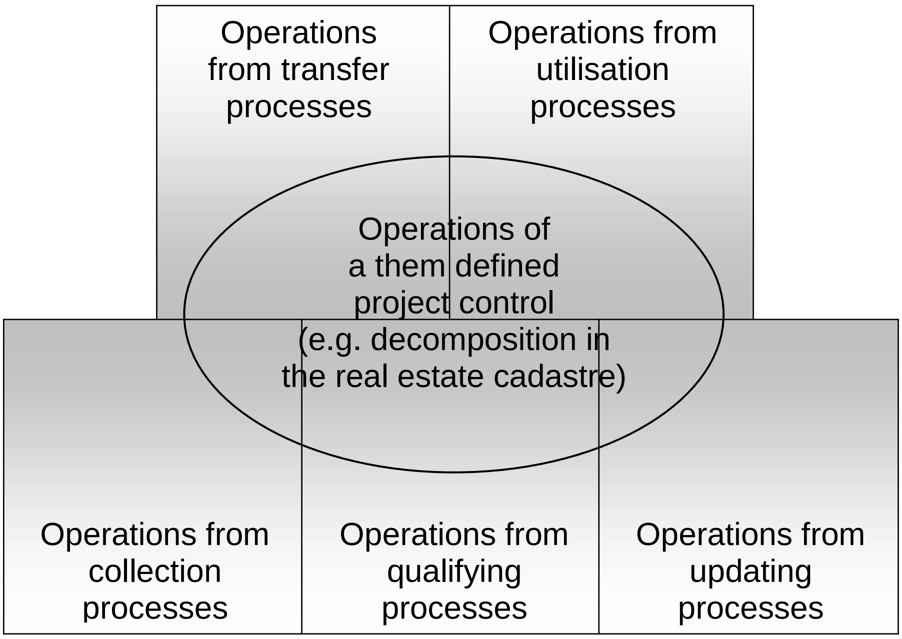
Figure 29: Processes and data for Geoinformation of Official Surveying and Mapping

One process consists of a hierarchy of several 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 for describing the selection and evaluation functionality for the creation of standard outputs (utilisation process).

3.8.2 Operation and Activity

For a complete application description, operations and activities that compose the data into functional dependencies and define the dynamic behaviour of the application must be defined. Operations are assigned to different processes of the AAA application schema. The following picture shows the dependencies between different processes.


Operations from transfer processes
Operations from utilisation processes
Operations of a them defined project control (e.g. decomposition in the real estate cadastre)
Operations from collection processes
Operations from qualifying processes
Operations from updating processes

An operation contains the representation of processing steps for qualification, management, use and transfer processes, each of which make reference 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 parameter). 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.

3.8.3 Processes of the AFIS-ALKIS-ATKIS application schema

By use of a process, a source database is transferred 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 that contain control parameters for the flow of processes, e.g. "AX_utilisation request" (AX_Benutzungsauftrag) in the utilisation process of the ALKIS application schema.

3.8.3.1 Data collection process

Source data is 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 is the object-structured collection data, which forms a basis for updating the official geoinformation.

3.8.3.2 Quality evaluation process

In the qualification evaluation process, the digital, object-structured collection data are transferred to updating data after 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 is the updating data.

3.8.3.3 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 is the primary database data.

The functionalities required for set-up and updating are described as part of the data exchange interface in 5.1.1, any other implicit functions of a management system are described in Section 5.3. The conceptual technical model for updating and the exact procedures during update processing are contained in the documentation on feature type "AX_revision request" (AX_Fortführungsauftrag). Further, a sequence diagram illustrates the description of the "AX_revision request".

3.8.3.4 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_K_updating_case_FM) 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 and usually carry the prefix “K” for complex data types (DE: komplexe Datentypen)

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.

Output feature types can also be presented in compliance with the portrayal catalogue, depending on requirement. The connections and information flow between the feature catalogue, the output catalogue, the portrayal catalogue and the outputs can be seen in the following schema illustration.

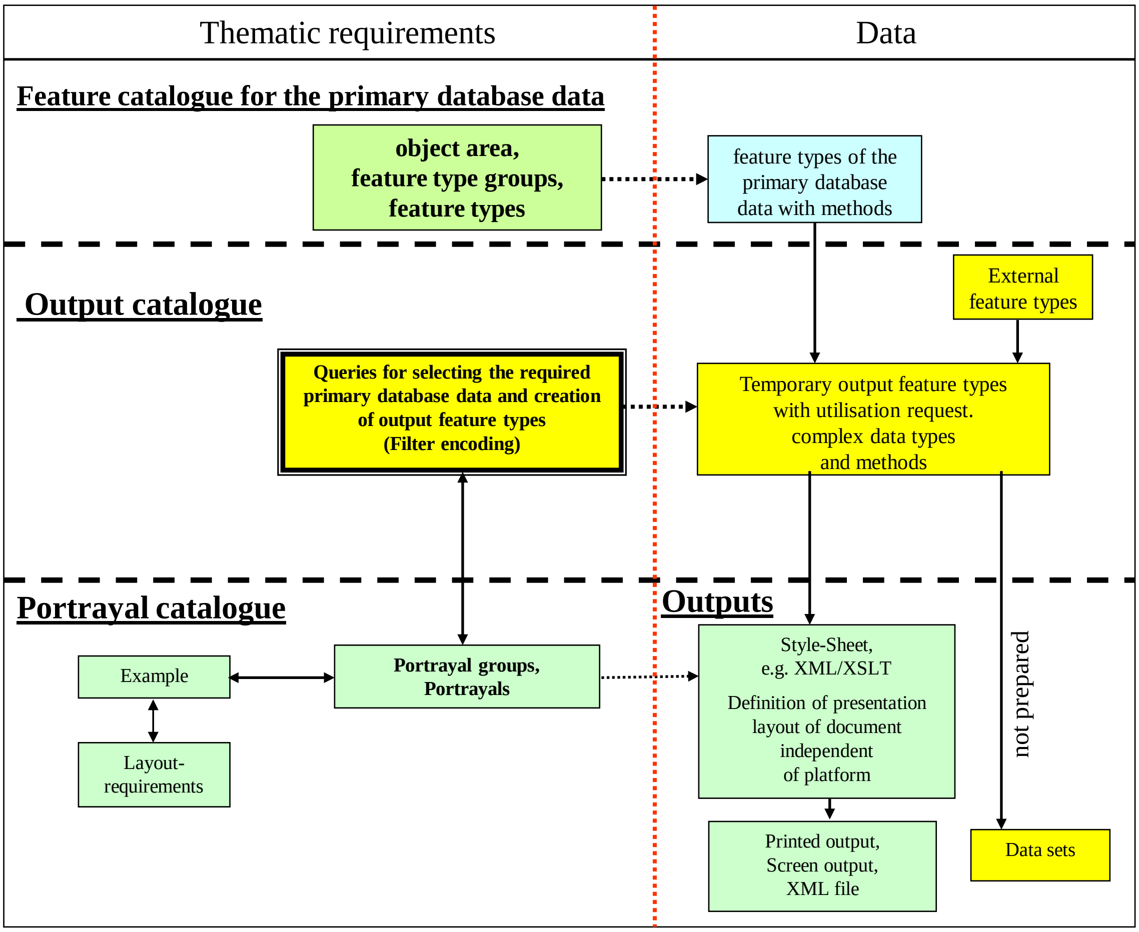

Thematic requirements
Data
Feature catalogue for the primary database data
object area, feature type groups, feature types
feature types of the primary database data with methods
Output catalogue
External feature types
Queries for selecting the required primary database data and creation of output feature types (Filter encoding)
Temporary output feature types with utilisation request. complex data types and methods
Portrayal catalogue
Outputs
Example
Portrayal groups, Portrayals
Layout-requirements
Style-Sheet, e.g. XML/XSLT
Definition of presentation layout of document independent of platform
not prepared
Printed output, Screen output, XML file
Data sets

Figure 31: ALKIS output schema

Accordingly, for a presentation output for example, the feature types of the primary database data are prepared using a defined enquiry sequence to develop a temporary output feature type. Taking into account the necessary portrayal and presentation layout, an output then appears on screen and/or is printed. It is also possible, however, to output unprepared output datasets that can be prepared by the users, using their own layout definitions.

3.8.3.5 Data transfer process

Transfer processes occur on the transfer of third party data in the form of updating data and on the submission of 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.

3.9 Project Control in the Basic Schema

The classes defined in the "AAA_Projectcontrol" (AAA_Projektsteuerung) describe a structural framework for describing a project control. The "AA_Request" (AA_Antrag), "AA_Projectcontrolcatalogue" (AA_Projektsteuerungskatalog) and "AA_Milestone" (AA_Meilestein) class diagrams illustrate the design of the project control modelled.

3.9.1 Request

The cornerstone of project control is the AA_Request feature type. This feature type implements a "mini-request-management", i.e an interface to external request management. In this way it is possible to generate a direct reference (with spatial reference) to an entry in the external request management (journal).

The request object also manages the resubmission of the request and assists the monitoring of the project control object. By means of the spatial reference, existing processes can be searched for in order to identify competing request or to consider neighbouring requests during the processing. The technical sequence of competing requests has to be defined by an administrator. The request object is connected to the project control object (AA_Projectcontrol (AA_Projektsteuerung)), in order to establish the assignment of the request to multiple project control objects and to check for combinations that are not allowed. Furthermore, the project control object controls and monitors the correct execution of the operations in the "technical qualification" sub-process. The update triggers are managed by the project control object.

The operation is a part of project control and is composed of individual activities. The operations themselves represent completed work steps. A predefined workflow defines the sequence and dependencies of the operations and their work steps. The operations are combined into groups and processed in a certain sequence, either in parallel or sequentially. The decision about the completion of the individual operations is documented in status (milestone).

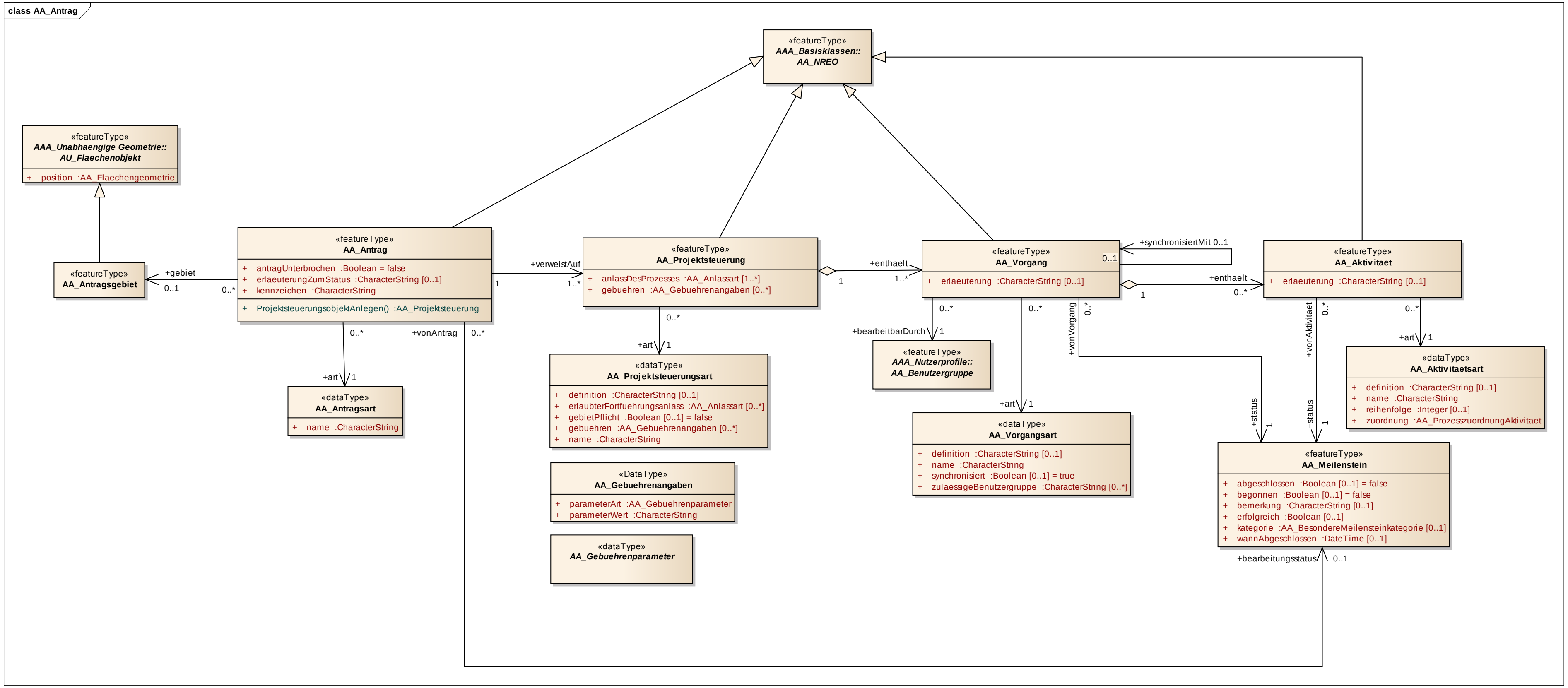

class AA_Antrag
«featureType»
AAA_Basisklassen::
AA_NREO
«featureType»
AAA_Unabhaengige Geometrie::
AU_Flaechenobjekt
+ position :AA_Flaechengeometrie
«featureType»
AA_Antragsgebiet
+gebiet
0..1
0..*
«featureType»
AA_Antrag
+ antragUnterbrochen :Boolean = false
+ erlaeuterungZumStatus :CharacterString [0..1]
+ kennzeichen :CharacterString
+ ProjektsteuerungsobjektAnlegen() :AA_Projektsteuerung
0..*
+vonAntrag
0..*
+art
1
«dataType»
AA_Antragsart
+ name :CharacterString
+verweistAuf
1
1..*
«featureType»
AA_Projektsteuerung
+ anlassDesProzesses :AA_Anlassart [1..*]
+ gebuehren :AA_Gebuehrenangaben [0..*]
0..*
+art
1
«dataType»
AA_Projektsteuerungsart
+ definition :CharacterString [0..1]
+ erlaubterFortfuehrungsanlass :AA_Anlassart [0..*]
+ gebietPflicht :Boolean [0..1] = false
+ gebuehren :AA_Gebuehrenangaben [0..*]
+ name :CharacterString
«DataType»
AA_Gebuehrenangaben
+ parameterArt :AA_Gebuehrenparameter
+ parameterWert :CharacterString
«dataType»
AA_Gebuehrenparameter
+enthaelt
1
1..*
«featureType»
AA_Vorgang
0..1
+ erlaeuterung :CharacterString [0..1]
+synchronisiertMit 0..1
+enthaelt
1
0..*
0..*
+bearbeitbarDurch
1
«featureType»
AAA_Nutzerprofile::
AA_Benutzergruppe
0..*
+art
1
«dataType»
AA_Vorgangsart
+ definition :CharacterString [0..1]
+ name :CharacterString
+ synchronisiert :Boolean [0..1] = true
+ zulaessigeBenutzergruppe :CharacterString [0..*]
+vonVorgang
0..*
«featureType»
AA_Aktivitaet
+ erlaeuterung :CharacterString [0..1]
+vonAktivitaet
0..*
0..*
+art
1
«dataType»
AA_Aktivitaetsart
+ definition :CharacterString [0..1]
+ name :CharacterString
+ reihenfolge :Integer [0..1]
+ zuordnung :AA_ProzesszuordnungAktivitaet
+status
1
+status
1
«featureType»
AA_Meilenstein
+ abgeschlossen :Boolean [0..1] = false
+ begonnen :Boolean [0..1] = false
+ bemerkung :CharacterString [0..1]
+ erfolgreich :Boolean [0..1]
+ kategorie :AA_BesondereMeilensteinkategorie [0..1]
+ wannAbgeschlossen :DateTime [0..1]
+bearbeitungsstatus
0..1

Figure 32: AA_Request (AA_Antrag) class diagram

3.9.2 Project control catalogue

Inside a project control object, the project control catalogue defines the types of update triggers allowed. It contains the project control types and operation types. The project control type combines the project control objects that exhibit a common characteristic. A similar concept applies to operation types.

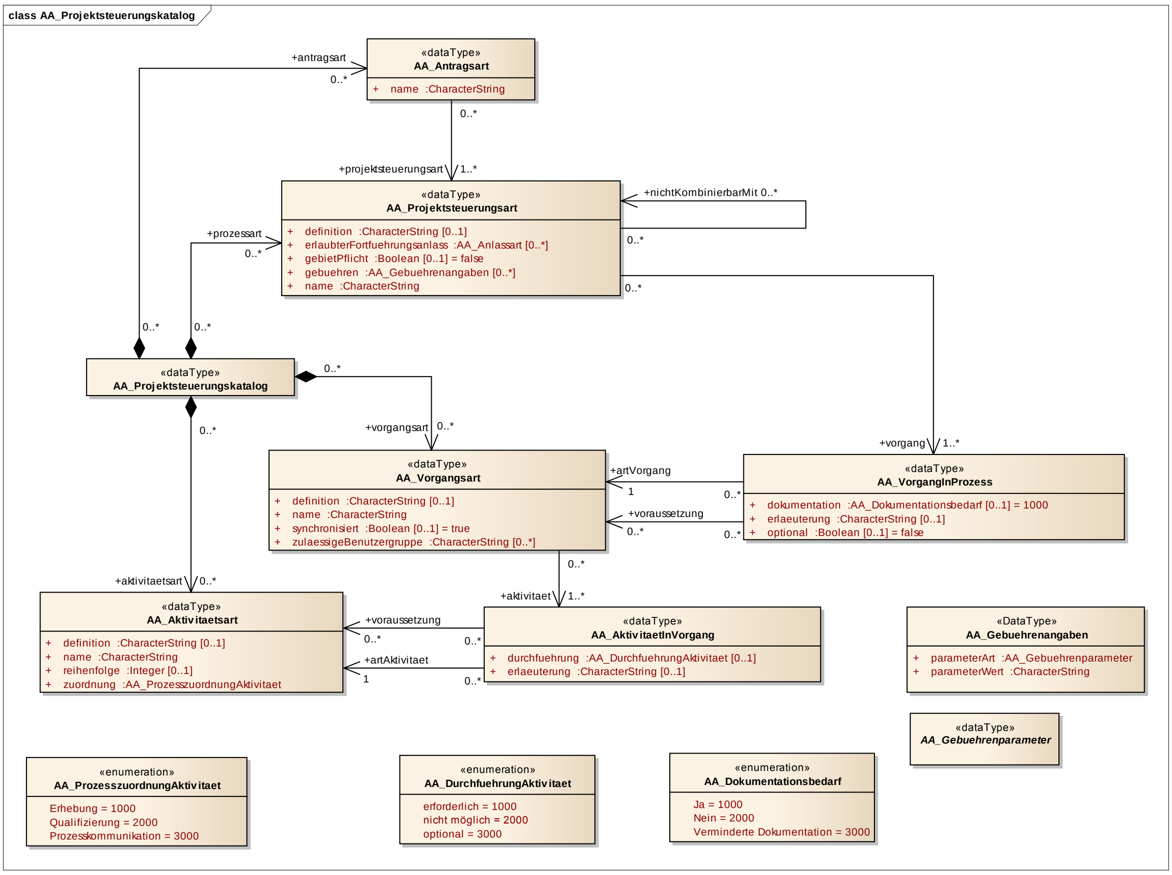

class AA_Projektsteuerungskatalog
+antragsart
0..*
«dataType»
AA_Antragsart
+ name :CharacterString
0..*
+projektsteuerungsart
1..*
«dataType»
AA_Projektsteuerungsart
+nichtKombinierbarMit 0..*
+prozessart
0..*
+ definition :CharacterString [0..1]
+ erlaubterFortfuehrungsanlass :AA_Anlassart [0..*]
+ gebietPflicht :Boolean [0..1] = false
+ gebuehren :AA_Gebuehrenangaben [0..*]
+ name :CharacterString
0..*
0..*
0..*
0..*
«dataType»
AA_Projektsteuerungskatalog
0..*
0..*
+vorgangsart
0..*
+vorgang
1..*
«dataType»
AA_Vorgangsart
+ definition :CharacterString [0..1]
+ name :CharacterString
+ synchronisiert :Boolean [0..1] = true
+ zulaessigeBenutzergruppe :CharacterString [0..*]
+artVorgang
1
0..*
+voraussetzung
0..*
0..*
«dataType»
AA_VorgangInProzess
+ dokumentation :AA_Dokumentationsbedarf [0..1] = 1000
+ erlaeuterung :CharacterString [0..1]
+ optional :Boolean [0..1] = false
0..*
+aktivitaetsart
0..*
+aktivitaet
1..*
«dataType»
AA_Aktivitaetsart
+ definition :CharacterString [0..1]
+ name :CharacterString
+ reihenfolge :Integer [0..1]
+ zuordnung :AA_ProzesszuordnungAktivitaet
+voraussetzung
0..*
0..*
+artAktivitaet
1
0..*
«dataType»
AA_AktivitaetInVorgang
+ durchfuehrung :AA_DurchfuehrungAktivitaet [0..1]
+ erlaeuterung :CharacterString [0..1]
«DataType»
AA_Gebuehrenangaben
+ parameterArt :AA_Gebuehrenparameter
+ parameterWert :CharacterString
«dataType»
AA_Gebuehrenparameter
«enumeration»
AA_ProzesszuordnungAktivitaet
Erhebung = 1000
Qualifizierung = 2000
Prozesskommunikation = 3000
«enumeration»
AA_DurchfuehrungAktivitaet
erforderlich = 1000
nicht möglich = 2000
optional = 3000
«enumeration»
AA_Dokumentationsbedarf
Ja = 1000
Nein = 2000
Verminderte Dokumentation = 3000

Figure 33: "AA_Projectcontrolcatalogue" (AA_Projektsteuerungskatalog) class diagram

3.9.3 Milestone

This is a data type that notes the current status and the responsibilities of an operation etc.

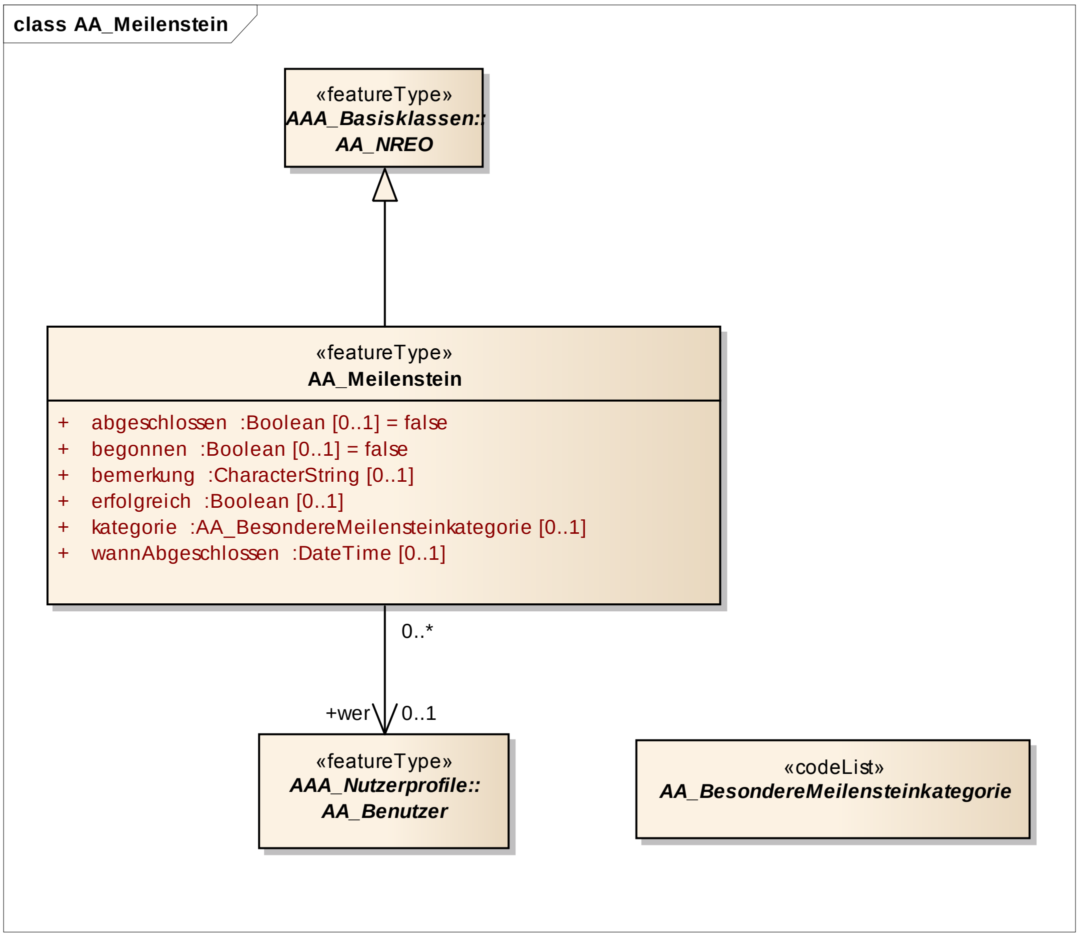

class AA_Meilenstein
«featureType»
AAA_Basisklassen::
AA_NREO
«featureType»
AA_Meilenstein
+ abgeschlossen :Boolean [0..1] = false
+ begonnen :Boolean [0..1] = false
+ bemerkung :CharacterString [0..1]
+ erfolgreich :Boolean [0..1]
+ kategorie :AA_BesondereMeilensteinkategorie [0..1]
+ wannAbgeschlossen :DateTime [0..1]
0..*
+wer
0..1
«featureType»
AAA_Nutzerprofile::
AA_Benutzer
«codeList»
AA_BesondereMeilensteinkategorie

Figure 34: "AA_Milestone" (AA_Meilenstein) class diagram

The GeoInfoDok "Further notes about ALKIS" (Erläuterungen zu ALKIS) contains an application-related explanation of the project control of the AAA-BasicSchema for the ALKIS technical schema. Further details are available there.

4 The Encoding of the NAS Schema

Chapter 2 describes the bases and correlations for the geoinformation to be described with this documentation. The reference model specified there 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-Archivewww.adv-online.de.

4.1 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,), 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.

4.2 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, currently with the *Enterprise Architect* software tool
- compliance with the regulations of ISO 19103 for the use of UML
- use of ISO 19107 (and therefore by implication ISO 19111), ISO 19115 and ISO 19123, and
- automated derivation and mapping of feature catalogues in accordance with ISO 19110.

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

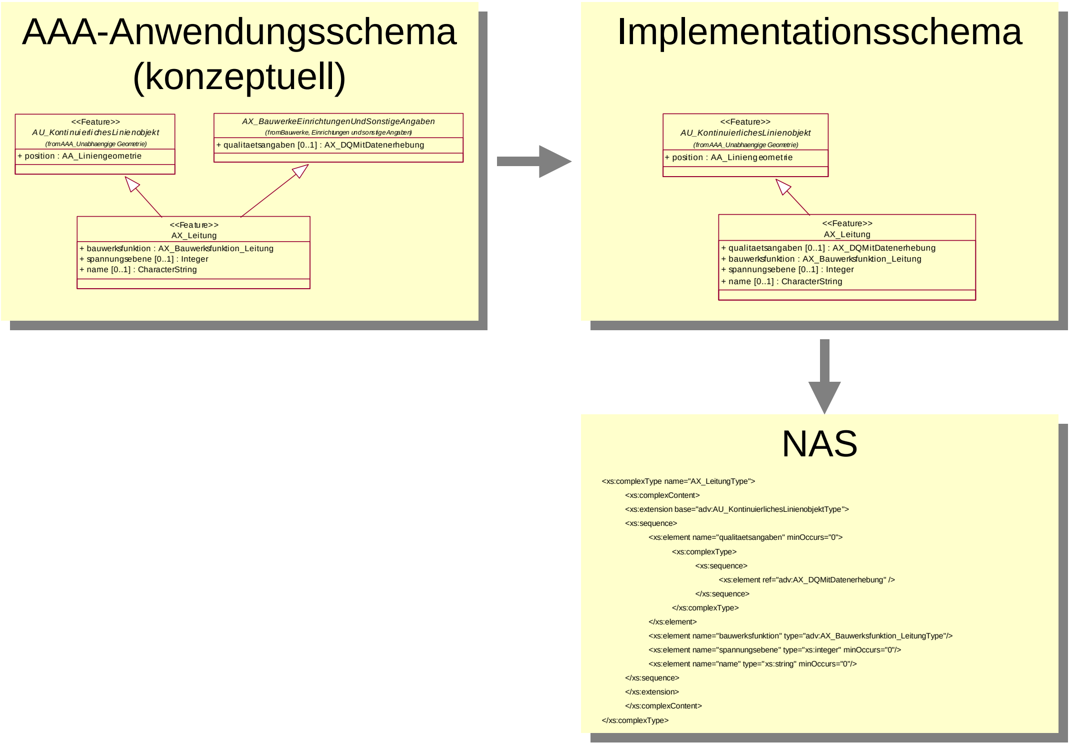
For this purpose, ISO 19118 *Encoding* defines a framework document for the creation of what are referred to as *Encoding Rules*, to derive interface definitions for data exchange from a UML application schema. The framework document for *Encoding Rules* defined in ISO 19118, Section 8 is applied for the NAS (Level-1 – Conformity with ISO 19118).

For the NAS a two level encoding process is applied (see following figure):

- In the first step, an implementation schema in UML for a GML application schema is derived, using scripts, from a conceptual, implementation-platform-independent AAA application schema. All elements of the implementation schema conform to the stipulations of ISO 19136 appendix E or - in the case of metadata elements - to ISO/TS 19139. ISO 19136 is identical to the *Geography Mark-up Language (GML)*OGC standard.

- In the second step the implementation schema is transferred, according to the encoding rules in ISO 19136 appendix E and – in the case of metadata elements – ISO/TS 19139, to a GML application schema.

Note also that the types, from ISO/TS 19103, ISO 19107, ISO 19111, ISO 19115 and ISO 19123 to ISO 19136 and ISO/TS 19139, used in the AAA application schema are present in standardised XML Schema implementations.


AAA-Anwendungsschema (konzeptuell)
<<Feature>>
AU_KontinuierlichesLinienobjekt
(fromAAA_Unabhaengige Geometrie)
+ position : AA_Liniengeometrie
AX_BauwerkeEinrichtungenUndSonstigeAngaben
(fromBauwerke, Einrichtungen und sonstige Angaben)
+ qualitaetsangaben [0..1] : AX_DQMitDatenerhebung
<<Feature>>
AX_Leitung
+ bauwerksfunktion : AX_Bauwerksfunktion_Leitung
+ spannungsebene [0..1] : Integer
+ name [0..1] : CharacterString
Implementationsschema
<<Feature>>
AU_KontinuierlichesLinienobjekt
(fromAAA_Unabhaengige Geometrie)
+ position : AA_Liniengeometrie
<<Feature>>
AX_Leitung
+ qualitaetsangaben [0..1] : AX_DQMitDatenerhebung
+ bauwerksfunktion : AX_Bauwerksfunktion_Leitung
+ spannungsebene [0..1] : Integer
+ name [0..1] : CharacterString
NAS

```
<xs:complexType name="AX_LeitungType">
    <xs:complexContent>
    <xs:extension base="adv:AU_KontinuierlichesLinienobjektType">
    <xs:sequence>
        <xs:element name="qualitaetsangaben" minOccurs="0">
            <xs:complexType>
                <xs:sequence>
                    <xs:element ref="adv:AX_DQMitDatenerhebung" />
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="bauwerksfunktion" type="adv:AX_Bauwerksfunktion_LeitungType"/>
        <xs:element name="spannungsebene" type="xs:integer" minOccurs="0"/>
        <xs:element name="name" type="xs:string" minOccurs="0"/>
    </xs:sequence>
    </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

Figure 35: Two level derivation process of the NAS

As well as the encoding of features, the AAA application and therefore the NAS also contains operations of a system for maintaining data in the primary database (update, initialisation, lock/unlock of objects, reservation of technical codes, request for output products including user-related updating of secondary database). Therefore the GML feature types are embedded within corresponding web-service-supportable operations, using elements of the OGC specifications Web Feature Service (WFS) and Filter Encoding (FES) that are complementary to GML. In this sense, an AAA-compliant database can be compared with an embedded Web Feature Server that additionally takes AAA-specific requirements into account.

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.

In the NAS the encoding of data corresponding to the specification of AdV-specific solutions is avoided as far as possible. Since the international standards need to be defined more precisely in some places for the special requirements of the AdV, the use of the Web

Feature Service and Filter Encoding is only possible with AFIS-ALKIS-ATKIS specific extensions

It is important to note that the platform-independent, conceptual model is fully described in the UML application schema. Future adaptations to the IT/GI mainstream will also become necessary for mapping onto specific implementation models (e.g. XML representations).

As well as standards from the ISO 19100 series, the following documents are used to define NAS:

- ISO/IEC 19501:2005, *Unified Modelling Language Specification (UML)*, http://www.uml.org/
- XML 1.0:1998, *Extensible Mark-up Language (XML)*, W3C Recommendation, 6 October 2000, http://www.w3.org/TR/2000/REC-xml-20001006
- XML Schema Part 1: *Structure*-W3C Recommendation, 2. Mai 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/
- XML Schema Part 2: *Structure*-W3C Recommendation, 2. Mai 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/
- Web Feature Service (WFS) 1.0
Open Geospatial Consortium, 2002
http://www.opengis.org/techno/specs/02-058.rtf
- Filter Encoding (FES) 1.0
Open Geospatial Consortium, 2002
http://www.opengis.org/techno/specs/02-059.rtf
- OWS Common Implementation Specification 1.0
Open Geospatial Consortium, 2005
https://portal.opengeospatial.org/files/?artifact_id=8798

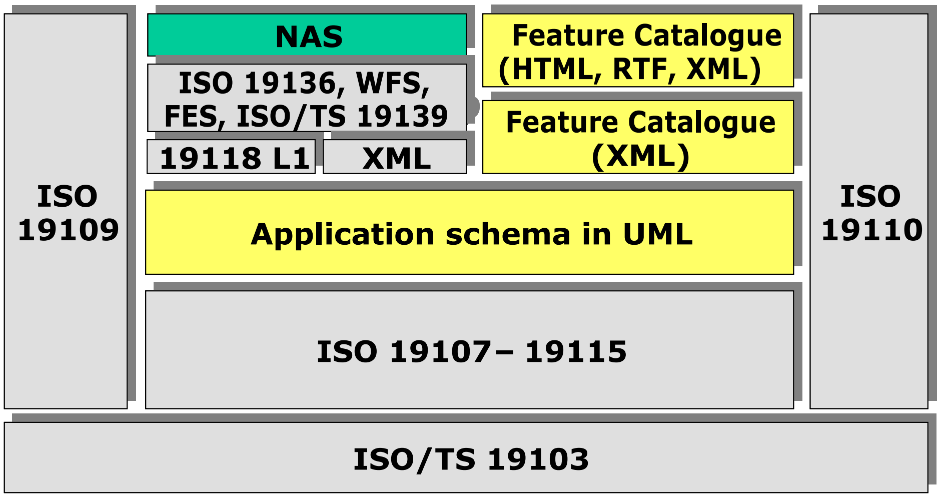

NAS
ISO 19136, WFS, FES, ISO/TS 19139
19118 L1
XML
Feature Catalogue (HTML, RTF, XML)
Feature Catalogue (XML)
ISO 19109
Application schema in UML
ISO 19110
ISO 19107– 19115
ISO/TS 19103

Figure 36: 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 an application schema, it is generally possible that generic GML software – so long as it has implemented the applied XML components - is able to conduct and syntactically interpret 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.

For this reason the capability requirements of software are specified and documented with the **GML profile** used by the NAS. During its definition, consideration was given to the objective that this profile should also cover application requirements over and above AFIS, ALKIS and ATKIS and evolve from an AdV-internal specification to a more widely accepted specification.

Due to the NAS being specified in terms of operations on data in a primary database and not purely as a "data format", in most cases the GML features in the NAS are embedded within the XML elements of operation calls and results. In the case of output of the primary database, for example, the set of GML features, i.e. the GML document, is embedded in the NAS results document and can be recognised and extracted very easily.

4.3 Encoding Process

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

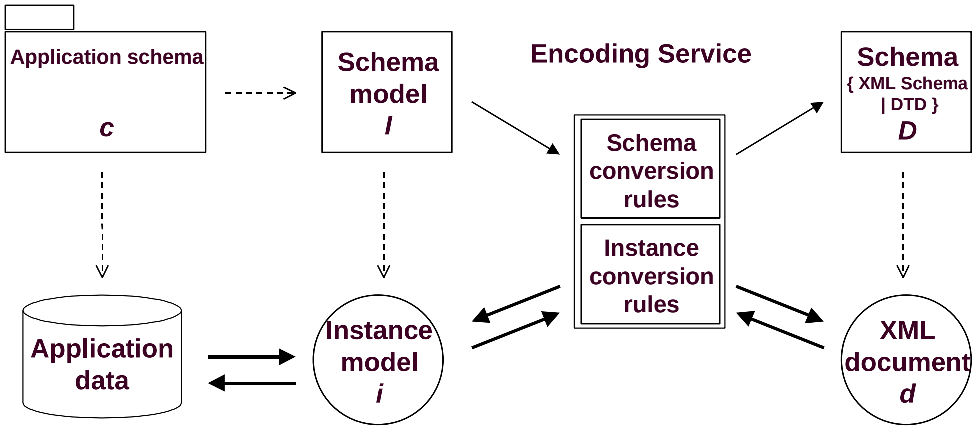


Figure 37: XML based encoding rules in accordance with ISO 19118

The process is based on the following general 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 schema based application data (object instances) is transferred to an XML file that has the same structure as the definitions of the XML Schema using *Instance Conversion Rules*.

Within the context of the NAS, the conversion of the AAA application schema (AAA-Anwendungsschema) (UML) in the GML application schema (XML Schema) is carried out with the following tools:

- execution of a software script inside the *Enterprise Architect* modelling program for conversion in the implementation schema
- export of the UML model in XMI 1.0 format with the XMI plug-in for *Enterprise Architect* developed by Unisys.
- application of encoding rules with the Open Source tool ShapeChange and creation of the NAS schema files.
  - The ShapeChange configuration file is adapted here to the AAA specific model general conditions.
  - As well as the XML Schema files of NAS, ShapeChange also creates the GML dictionary files of the classes of the UML model with all definitions including value types.

4.4 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 *Enterprise Architect*) 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 (*Austausch-Metadaten)*
  4. Identifiers
  5. Update mechanism
- Input data structure
- Output data structure
- Presentation rules
- Example

4.4.1 Requirements

Application schema

The AAA application schema was developed based on the rules for application schemas from ISO/TS 19103 and ISO 19109.

In order to increase clarity the following additional stereotypes were used in the application schema.

- <<FeatureType>> in terms of the definition in ISO 19136 appendix E,
- <<Request>> and <<Response>> for the messages exchanged during the execution of NAS operations.

Additionally UML Tagged Values, as specified in ISO 19136 E.2.1, are used in the model, and two further UML Tagged Values "xsdEncodingRule and "reverseRoleNAS" supported. For this, the following rules apply::

- version, targetNamespace and xmlns: current values according to the version of the AAA application schema, only in AAA application schema package
- gmlProfileSchema: Pointers to the GML profile file, only in AAA application schema package
- xsdDocument: filename of the XML Schema file is set for the AAA application schema package as well as for the AAA basic schema, AAA technical schema and NAS operations packages.
- with classes, the following UML Tagged Values are set:
  - noPropertyType: "true" with <<FeatureType>>, <<Request>> and <<Response>>; "false" with <<DataType>> and <<Union>>
  - byValuePropertyType: "false" with <<FeatureType>>, <<Request>>, <<Response>>,<<DataType>> and <<Union>>
  - isCollection: "false" with <<FeatureType>>, <<Request>>, <<Response>>,<<DataType>> and <<Union>>
  - asDictionary: "true", only with <<CodeList>>:
- With attributes and association roles the following Tagged Values are set:
  - sequenceNumber
  - inlineOrByReference: "byReference" with <<FeatureType>>-valued property, otherwise "inline"
  - isMetadata: "true" with all quality information, otherwise false; all types that are prefixed with the following character strings are counted for quality information: "LI_", "DQ_", "AX_DQ", "AX_LI"

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 repertoire. This is identical to the *Unicode Character Repertoire*.

"UTF-8" (*UTF = UCS Transformation Format*) should be used uniformly 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 exchange metadata required is modelled and transferred by automatic conversion to 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 overlooked and ignored.

The identifiers of 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 3.3.10.

Update mechanism

An update mechanism as defined by ISO 19118, Section 8 is supported via NAS operations.

4.4.2 Input data structure

The AAA application schema uses some constructions in UML that are not supported in the encoding rules of ISO 19136 appendix E and ISO/TS 19139. Thus there is a script supported transformation of the conceptual AAA application schema in UML in an implementation schema (see above).

The script makes the following changes.

- Multiple inheritance: while neither ISO 19136 nor ISO/TS 19139 support multiple inheritance in the encoding rules, the AAA model uses these. However, in Mixin classes (e.g. AP_GPO, AX_Catalogueentry (AX_Katalogeintrag) the Mixin classes are deleted:

  - all attributes are copied into the next in the NAS coded subtypes.
  - all relations to the Mixin classes are also copied, respectively, to the next in the NAS coded subtypes. In this way the role name is changed through suffixing the class name, in order to ensure that property names are unambiguous, and
  - the <<Type>> classes are deleted.

- Un-navigable association roles are set to
  - navigable
  - so long as not present with the name "inversto_" and the name of the inverse roll present
  - present with a minimum cardinality of "0", and
  - the UML Tagged Value "reverseRoleNAS" is set to "true".

- Those model elements that have contents that are not converted into the NAS, are removed during the derivation of the implementation model for the data exchange.
  - Package:
    - "AAA Versioningschema" (AAA Versionierungsschema
  - Attributes:
    - "AA_Object.identifier" (AA_Objekt.identifikator)
  - Classes:
    - "AA_ObjectWithoutSpatialReference" (AA_ObjektOhneRaumbezug)
    - "AX_Revision" (AX_Fortfuehrung)
    - "AX_Database" (AX_Datenbank)
    - "AX_Operation_Database" (AX_Operation_Datenbank)
    - "AX_TemporaryArea" (AX_TemporaererBereich)
    - "AX_NewObject" (AX_NeuesObjekt)
    - "AX_DeletedObject" (AX_GeloeschtesObjekt)
    - "AX_UpdatesObject" (AX_AktualisiertesObjekt)
    - "AX_RevisionObject" (AX_Fortfuehrungsobjekt)

- Those model elements that have contents that are converted in a specific manner into the NAS, are adapted accordingly.
  - With Mixin classes, the properties of AA_PMO and AA_Object are transferred to "AD_PointCoverage" (AD_PunktCoverage) and "AD_GridCoverage" (AD_GitterCoverage) and the conceptual attributes deleted. In addition the inheritance relationships are set to "CV_DiscreteGridPointCoverage" or "CV_DiscretePoint Coverage".

- In order to minimise the size of the model for the NAS derivation process and because the UML from the ISO packages generates numerous errors and warnings, the ISO packages are deleted where possible.
- The following types receive a new attribute:
  - "TA_PointComponent.position : "GM_Point"
  - "TA_CurveComponent.position : "GM_Curve"
  - "TA_SurfaceComponent.position : GM_Surface"
  - "TA_MultiSurfaceComponent.position : GM_Object" (the value must be either GM_Surface or GM_MultiSurface)
- The following attributes receive a new type:
  - "AU_Multipointobject.position (AU_Punkthaufenobjekt.position) : GM_MultiPoint"
  - "AU_ContinuousLinesobject.position (AU_KontinuierlichesLinienobjekt.position) : GM_Curve"
  - "AU_Surfaceobject.position (AU_Flaechenobjekt.position) : GM_Object"
  - "AG_Surfaceobject.position (AG_Flaechenobjekt.position) : GM_Object"
  - "AG_Pointobject.position (AG_Punktobjekt.position) : GM_Point"
  - "AU_Object.position" (AU_Objekt.position) : GM_Object"
  - "AG_Object.position" (AG_Objekt.position) : GM_Object"
  - "AU_GeometryObject_3D.position" (AU_GeometrieObjekt_3D.position) : GM_Object"
  - "AU_MultipleLinesObject_3D.position" (AU_MehrfachLinienObjekt_3D.position) : GM_Object"
  - "AU_MultipleSurfacesObject_3D.position" (AU_MehrfachFlaechenObjekt_3D.position) : GM_Object"
  - "AU_RingObjekt_3D.position : (AU_UmringObjekt_3D.position) : GM_MultiPoint"
  - "AU_Multipointobject_3D.position (AU_PunkthaufenObjekt_3D.position) : GM_MultiPoint"
  - "AX_DQWithoutDatasurvey.origin [0..1] (OhneDatenerhebung.herkunft [0..1]) : LI_Lineage"
  - "AX_DQWithDatasurvey.origin [0..1] (AX_DQMitDatenerhebung.herkunft [0..1]) : LI_Lineage"
  - "AX_DQPointlocation.origin [0..1] (AX_DQPunktort.herkunft [0..1]) : LI_Lineage"

  - "AX_Severeanomaly_Severity.value (AX_Schwereanomalie_Schwere.wert) : Measure"
  - "AX_Blockorder.uuidList [1..*] (AX_Sperrauftrag.uuidListe [1..*]) : URI"
  - "AX_Unblockorder.uuidList [1..*] (AX_Entperrauftrag.uuidListe [1..*]) URI"
  - "ExceptionUpdating.Objectalreadyblocked [0..*] (ExceptionFortfuehrung.bereitsGesperrteObjekte [0..*]) : URI"
  - "ExceptionUpdating.Objectnotuptodate [0..*] (ExceptionFortfuehrung.nichtMehrAktuelleObjekte [0..*]) : URI"
  - "ExceptionAAAFortfuehrungOderSperrung.bereitsGesperrteObjekte [0..*] : URI"
  - "ExceptionAAAFortfuehrungOderSperrung.nichtMehrAktuelleObjekte [0..*] : URI"
  - "ExceptionAAAEntsperren.uuidListe [0..*] : URI"
  - "DCP.HTTP : URI"
  - "DCP.email : URI"

- Applications with collections as values, type and cardinality are adapted:
  - "AA_Objekt.modellart [1..*] : AA_Modellart"
  - "AA_Objekt.anlass [0..2] : AA_Anlassart"
  - "AA_Objekt.zeigtAufExternes [0..*] : AA_Fachdatenverbindung"
- Pointers into the project control catalogues are implemented as GML dictionary pointers:
  - "AA_Antrag.art : GenericName"
  - "AA_Projektsteuerung.art : GenericName"
  - "AA_Vorgang.art : GenericName"
  - "AA_Aktivitaet.art : GenericName"
- Classes of the project control catalogue are deleted:
  - "AA_Antragsart"
  - "AA_Projektsteuerungsart"
  - "AA_Vorgangsart"
  - "AA_Aktivitaetsart"
  - "AA_Projektsteuerungskatalog"
  - "AA_AktivitaetInVorgang"
  - "AA_VorgangInProzess"
  - "AA_Dokumentationsbedarf"
  - "AA_DurchfuehrungAktivitaet"
  - "AA_ProzesszuordnungAktivitaet"

- As a result of the adaptations above, all except the following types are deleted:
    - "AA_Liniengeometrie"
    - "AA_Flaechengeometrie"
    - "AU_Geometrie"
    - "AG_Geometrie"
    - "AU_Geometrie_3D"
    - "AA_Punktgeometrie"
    - "AA_Punktgeometrie_3D"
    - "AA_MultipleLinesGeometry_3D" (AA_MehrfachLinienGeometry_3D) :
    - "AA_MultipleSurfacesGeometry_3D" (AA_MehrfachFlaechenGeometry_3D) :
    - "AA_PunktLinienThema"
    - "TS_PointComponent":
    - "TS_CurveComponent"
    - "TS_SurfaceComponent"
    - "TS_Feature"
    - "AX_LI_Lineage_OhneDatenerhebung"
    - "AX_LI_Lineage_MitDatenerhebung"
    - "AX_LI_Lineage_Punktort"
    - "AX_LI_ProcessStep_OhneDatenerhebung"
    - "AX_LI_ProcessStep_MitDatenerhebung"
    - "AX_LI_ProcessStep_Punktort"
    - "AX_LI_ProcessStep_OhneDatenerhebung_Description"
    - "AX_LI_ProcessStep_MitDatenerhebung_Description"
    - "AX_LI_ProcessStep_Punktort_Description"
    - "AX_LI_Source_MitDatenerhebung"
    - "AX_LI_Source_Punktort"
    - "Acceleration"
    - "AD_ReferencableGrid" (AD_ReferenzierbaresGitter)
    - "AD_Valuematrix" (AD_Wertematrix"
    - "AA_UUID"
- For all classes the UML Tagged Value "xsdEncodingRule" is set: "iso19136_2007", except for types that begin with the character string "LI_", "DQ_", "AX_DQ", "AX_LI", for these "iso19139_2007" is used.

The instance schema is adopted based on the ISO 19136 E.2.2 implementation schema.

4.4.3 Output data structure

The output data structure is explained under ISO 19136 E.2.3.

4.4.4 Schema encoding rules

Relevant sources of the schema encoding rules include ISO 19136 E.2.4 and, for classes with the UML Tagged Value xsdEncodingfRule with a value "iso19139_2007", ISO/TS 19139 chapter 8

The values of the UML Tagged Value "reverseRoleNAS" are output in XML Schema in appinfo-annotations, in the element that corresponds to the association.

The schema, specified by AdV, for WFS extensions codes the extensions described in section 5.1.3.

When importing schemas defined and managed by third parties (OWS Common 1.0.0, GML 3.2.1, Xlink 1.0.0, ISO/TS 19139:2007, ShapeChange 1.0, WFS 1.0.0 und Filter Encoding 1.0.0) they are copied to a local directory and adapted to the local file structure. The schemas of WFS and Filter Encoding import into the NAS GML 3.2.1 (instead of GML 2.1.2 as in standardised schemas), therefore the namespaces of these schemas are also changed.

Encoding rules for instances

This chapter describes the representation of the instance model in corresponding XML elements. The result of the mapping 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 set.

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

Encoding of identifiers in the NAS

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

In the NAS the AAA identifier is to be encoded in the XML attribute gml:id. Example:

```
<AX_Gebaeude gml:id="DEST123412345678">
   <!-- ... -->
</AX_Gebaeude>
```

In order to ensure the documents-wide uniqueness of gml:id attributes, the information is always supplemented with creation date/time when multiple versions of an object in an XML document are present. This happens in particular in the following cases:

- Multiple version of an object are selected in a primary database data output.
- An object is changed multiple times with multiple updating cases in an update request.
- In the set-up, historical object versions are migrated.

Date and time are encoded without separation characters, in order that they fulfil XML ID conditions, i.e. in the following form: CCYYMMDDThhmmssZ.

Example:

```
<AX_Gebaeude gml:id="DEST12341234567820010101T110000Z">
   <!-- ... -->
</AX_Gebaeude>

<!-- ... -->

<AX_Gebaeude gml:id="DEST12341234567820070313T125420Z">
   <!-- ... -->
</AX_Gebaeude>
```

In addition to the `gml:id`, the identifier is also to be encoded in the object property `gml:identifier`, predefined in ISO 19136. For this, the following rules apply:

- "http://www.adv-online.de/" is to be used as codeSpace.
- The identifier is always to be given (but without creation date/time).
- The identifier is encoded as a global identifier, i.e. as URN (see below).

Example:

```
<AX_Gebaeude gml:id="DEST123412345678">
   <gml:identifier codeSpace="http://www.adv-online.de/">
      urn:adv:oid:DEST123412345678
   </gml:identifier>
   <!-- ... -->
</AX_Gebaeude>
```

In the NAS there are two types of references to objects

- References from one object to another are always represented as XLink. Inside the NAS, references to other AAA objects are to be expressed with URNs, without exception. Uniform Resource Names (URNs) are globally unique, persistent, storage location independent identifiers. URNs of AAA identifiers are all prefixed with urn:adv.oid, supplemented by the identifier. Example: "urn:adv:oid:DEST123412345678".
- References from a selection criterion to a specific object via an identifier (ogc:FeatureId/@fid). Here the identifier is always given without URN context. In some cases, here in the case of actuality checking, the 16 position creation date / time without separators is to be given. The corresponding cases are specified in section 5.1.

Encoding of geometry properties in the NAS

The orientation of lines (curves) is not encoded in the AAA basic schema. Because the direction of features (e.g. flow direction of rivers etc.) is sometimes of interest, it must be ensured that a) the collection is in a positive direction and b) this direction is maintained during the course of processing and storage. Thus it can always be assumed that 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 positive-direction-oriented boundary 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 - if they are interested in that 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 jointly used geometry can be assigned to themes. Topological relationships and common geometric usage are possible only within a theme. A theme is always 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.

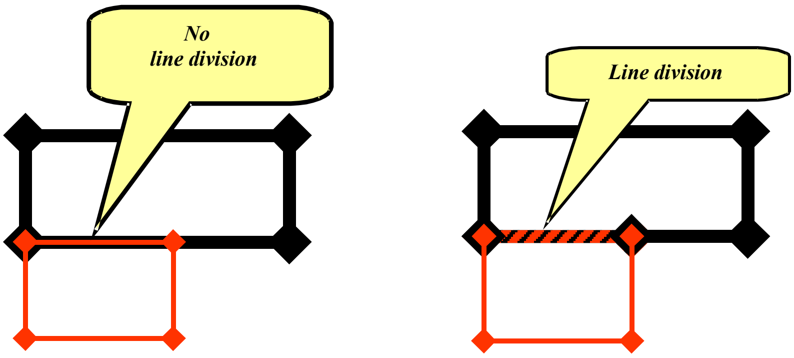


Figure 38: Explanation of line division

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 interpolation type used. A sequence reversal is also permitted.

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

For clarification the two situations in figure 42 are encoded as examples. First the left situation (no line division):

```
...
<AX_Flurstueck gml:id="DEBY000000000001">
  <!-- ... -->
  <position>

    <gml:Surface srsName="urn:adv:crs:DE_DHDN_3GK3" gml:id="_1">
      <gml:patches>
        <gml:PolygonPatch>
          <gml:exterior>
            <gml:Ring gml:id="_2">
              <gml:curveMember>
                <gml:Curve gml:id="_3">
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:posList>601085.954 5943996.138</gml:posList>
                      <gml:posList>601085.954 5943998.138</gml:posList>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve gml:id="_4">
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:posList>601085.954 5943998.138</gml:posList>
                      <gml:posList>601078.954 5943998.138</gml:posList>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve gml:id="_5">
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:posList>601078.954 5943998.138</gml:posList>
                      <gml:posList>601078.954 5943996.138</gml:posList>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve gml:id="_6">
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:posList>601078.954 5943996.138</gml:posList>
                      <gml:posList>601085.954 5943996.138</gml:posList>
                    </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 srsName="urn:adv:crs:DE_DHDN_3GK3" gml:id="_7">
      <gml:patches>
        <gml:PolygonPatch>
          <gml:exterior>
            <gml:Ring gml:id="_8">
              <gml:curveMember>
                <gml:Curve gml:id="_9">
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:posList>601082.954 5943994.138</gml:posList>
                      <gml:posList>601082.954 5943996.138</gml:posList>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve gml:id="_10">
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:posList>601082.954 5943996.138</gml:posList>
                      <gml:posList>601078.954 5943996.138</gml:posList>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve gml:id="_11">
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:posList>601078.954 5943996.138</gml:posList>
                      <gml:posList>601078.954 5943994.138</gml:posList>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve gml:id="_12">
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:posList>601078.954 5943994.138</gml:posList>
                      <gml:posList>601082.954 5943994.138</gml:posList>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
            </gml:Ring>
          </gml:exterior>
        </gml:PolygonPatch>
      </gml:patches>
    </gml:Surface>
  </position>
  <!-- ... -->
</AX_Gebaeude>
...
```

For comparison now the situation on the right, where the lower edge of the plot is divided into two edges in order to achieve geometry identity in the sense described above:

```
...
<verwendeteInstanzenthemen>cadastral parcels and
buildings</verwendeteInstanzenthemen>
...
```

```
<AX_Flurstueck gml:id="DEBY000000000001">
  <!-- ... -->
  <position>
    <gml:Surface srsName="urn:adv:crs:DE_DHDN_3GK3" gml:id="_1">
      <gml:patches>
        <gml:PolygonPatch>
          <gml:exterior>
            <gml:Ring gml:id="_2">
              <gml:curveMember>
                <gml:Curve gml:id="_3">
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:posList>601085.954 5943996.138</gml:posList>
                      <gml:posList>601085.954 5943998.138</gml:posList>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve gml:id="_4">
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:posList>601085.954 5943998.138</gml:posList>
                      <gml:posList>601078.954 5943998.138</gml:posList>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve gml:id="_5">
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:posList>601078.954 5943998.138</gml:posList>
                      <gml:posList>601078.954 5943996.138</gml:posList>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve gml:id="_6">
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:posList>601078.954 5943996.138</gml:posList>
                      <gml:posList>601082.954 5943996.138</gml:posList>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve gml:id="_7">
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:posList>601082.954 5943996.138</gml:posList>
                      <gml:posList>601085.954 5943996.138</gml:posList>
                    </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 srsName="urn:adv:crs:DE_DHDN_3GK3" gml:id="_8">
      <gml:patches>
        <gml:PolygonPatch>
          <gml:exterior>
            <gml:Ring gml:id="_9">
              <gml:curveMember>
                <gml:Curve gml:id="_10">
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:posList>601082.954 5943994.138</gml:posList>
                      <gml:posList>601082.954 5943996.138</gml:posList>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve gml:id="_11">
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:posList>601082.954 5943996.138</gml:posList>
                      <gml:posList>601078.954 5943996.138</gml:posList>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve gml:id="_12">
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:posList>601078.954 5943996.138</gml:posList>
                      <gml:posList>601078.954 5943994.138</gml:posList>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
              <gml:curveMember>
                <gml:Curve gml:id="_13">
                  <gml:segments>
                    <gml:LineStringSegment>
                      <gml:posList>601078.954 5943994.138</gml:posList>
                      <gml:posList>601082.954 5943994.138</gml:posList>
                    </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 they can only occur in a complete form (type of theme declaration = "all objects", class themes).
- For the complete, class-related form all features of a feature class belong automatically to this theme. There is no choice.

The instance-related specification of the geometric identity that was previously possible up until version 6.0.1 is no longer provided for since an unambiguous correlation between a geometric identity and the corresponding theme was not possible but rather could only be stated universally for all instances in the product metadata. Furthermore, no technical requirements were known for capturing and managing this functionality.

Encoding of coordinate reference systems in NAS

In principle, each geometry unit in the NAS file (point, line, surface) must refer to a coordinate reference system (CRS). This can be accomplished either implicitly by specification of the CRS in the case of higher level geometry entity or explicitly by the respective geometry entity. The reference is created by giving a URI *(Uniform Resource Identifier)*. In order to avoid having to give this URI for each and every feature geometry, all reference systems used, one of which can be identified as a standard reference system, are given 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 used for all geometries not present in the standard reference system. The syntax described in chapter 7 and the designations defined there have to be used.

For NAS documents that contain objects in a "FeatureCollection", the standard reference system has to be stated in the attribute "srsName" of "gml:envelope".

Furthermore, the declaration of the coordinate reference system used in the exchange metadata indicates the coordinate resolution 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 location coordinates is specified at 3 decimal places (mm). It is necessary to indicate the relevant number of decimal places, as neither GML nor ISO 19107 *Spatial Schema* make any restriction or provide any options in this respect (data type: *decimal* or *double*). The following definition is 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 references to units of measure in NAS

Basically, each value in the NAS file representing a unit of measurement (e.g. length, area, angle) must have a reference to a unit of measurement. This reference is made by specifying a URI *(Uniform Resource Identifier)* in the "uom" attribute. The syntax described in chapter 8 and the designations defined there have to be used.

4.5 GML-Profile for the 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 "greys out" those parts of GML not used in the actual version (such as topology or unused object properties).

Besides omitting the un-needed GML structures, a number of decisions were made on the use of GML in NAS. The objective is to restrict freedom on the kind of encoding, so as to simplify the use of NAS:

1. For GML-objects which allowed the use of both normal and array properties, one of the variants, usually the array properties, was deleted.

2. In the portrayal of GM_Curve in AFIS, ALKIS and ATKIS data, gml:Curve may only be used with exactly one gml:LineStringSegment if the GM_Curve has linear interpolation in all segments. (gml.LineString must not be used in these cases and its application is allowed solely in Filter Encoding expressions).

3. In the portrayal of GM_PolyhedralSurface in AFIS, ALKIS and ATKIS data, gml:surface may only be used with exactly one gml:PolygonPatch. (gml.Polygon must not be used in these cases and its application is allowed solely in Filter Encoding expressions).

4. As the majority of area-objects (e.g. parcels), which are defined as GM_MultiSurface consist only of a single area, gml:_AbstractSurface is always to be used if there is only a single area and gml:MultiSurface is allowed only if there are two or more separated areas.

5. For the encoding of GM_Ring in AFIS, ALKIS and ATKI data gml:Ring with exactly one gml:Curve must be used if the GM_Ring consists of only one single GM_Curve. (gml.LinearRing must not be used in these cases and its application is allowed solely in Filter Encoding expressions).

6. For coordinate data, gml:pos (with gml:Point) or gml:posList (with other geometry objects) must be used.

7. The standard properties of GML features "gml:name" and "gml:description" may only be used in GML dictionaries, but not for properties of features in the namespace of the NAS.

Note that the schemas file of the profile (gmlProfileNAS.xsd) is not used for validation by defaut (because of the rule, as in XML Schema the schemaLocation-Attribute is interpreted). In order to rule out the possibility of the wrong interpretation, the namespace of the schema file of the GML profile is also changed to a namespace other than from XML Schema. If the schema file is to be used locally for validation purposes, the data content must be checked accordingly.

5 NAS Operations

In principle, NAS is designed firstly for communication to the "outside", i.e. for users of AAA data. Depending on the implementation concept, it can also be used for the internal communication between recording or qualification systems and management systems. Those functionalities are also considered in the following chapters. An implementation that enables internal communication with system-specific functions only has to provide those NAS operations, from the range described, that are relevant to the output of data to third parties. This includes, in particular, the 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.

A uniform and standardised arrangement for the transmission of data between the systems of different AAA software providers is to be realised at a Federal State specific level. Due to the differing Federal State solutions, a uniform regulation in the GeoInfoDoc regarding automated communication (e.g. with SOAP) between primary and secondary data storage is not possible.

For usage in thematic information systems, three general operations are specified for updating the primary database, for requesting outputs of that data and for enquiring generally about the properties of primary data base storage.

5.1 Scope of Functions

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

- Setting-up and Updating a Primary Database
- Request for outputs
  - Output of utilisation data (extracts)
  - Management of secondary database (initial tagging and updating)
- Locking and Unlocking Objects
- Reservation (of point numbers etc.)
- Transfer of protocol information
  (e.g. processing protocols, error protocols)
- Determining the Properties of a Primary Data Storage

Each NAS operation comprises two XML schema definitions, one for requesting the operation (*Request*) and one for the response (*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 multiple operations is allowed. 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 derived automatically from UML models. For the AFIS and ALKIS applications, the UML models have already been created. 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 schemas for the NAS operations are summarised in the file NAS-Operationen.xsd. The generally usable basic operations are contained in the file AAA-Basisschema.xsd.

The operations are based on the OWS Common Implementation Specification 1.0 that must be adhered to. In particular, each NAS implementation must support the GetCapabilities operation.

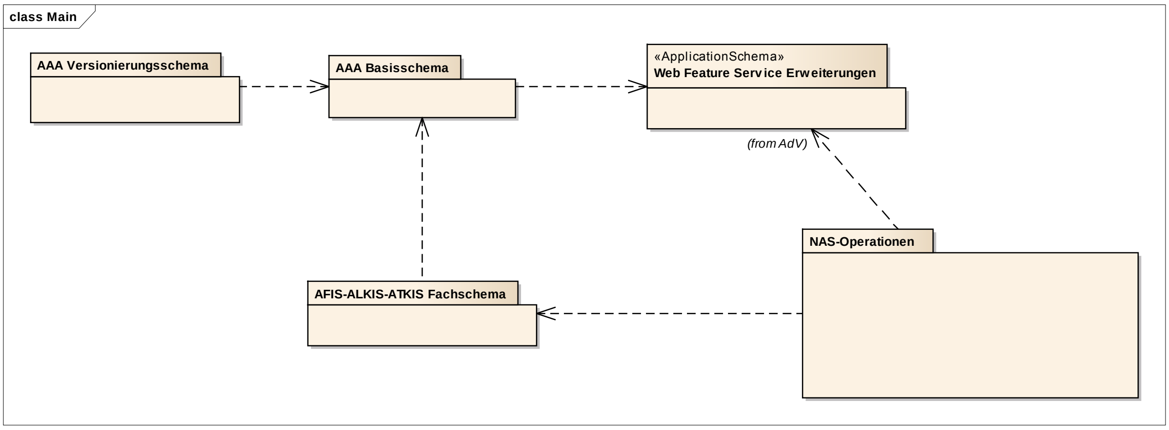


Figure 39: UML package "NAS-operations" in the context of application schema components

5.1.1 Setting-up and Updating a Primary Database

Because GML itself provides no elements for updating operations, the definitions from the OGC's *Web Feature Service* (WFS) are used for this purpose. As well as the transaction mechanism, the WFS specification defines 3 change functions: <Insert> (insert new object), <Replace> (amend, overwrite object) and <Delete> (delete object). Which changes result in <Replace> or <Delete> followed by <Insert> is to be specified in the features catalogue. 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>

  <adv:Replace typeName="AX_Flurstueck">
    <AX_Flurstueck gml:id="DEBY0000F000000220010101T000000Z">
      ...
    </AX_Flurstueck>
    <ogc:Filter>
      <ogc:FeatureId fid="DEBY0000F000000220010101T000000Z"/>
    </ogc:Filter>
  </adv:Replace>

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

Note the following:

- The filter expressions for <Delete> and <Replace> operations may contain only *FeatureId* elements. More complex filter criteria are not allowed.
- With *FeatureId* conditions in "<Replace>" and "<Delete>" operations, the identifier must be supplemented with creation date and time, in order to be able to test that it is up-to-date (actuality test). Date and time are encoded without separation characters, in order that they fulfil XML ID conditions, i.e. in the following form: CCYYMMDDThhmmssZ. For the extension to the creation date/time the following exception applies: 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 onwards. Only in this special case is there no testing of actuality and the latest – just created – version is used.

- In <Delete>-operations it is only possible to handle more than one AAA-Object if they all belong to the same feature class; in <Insert> operations there can be different feature types. <Replace> operations always deal with only one object.
- <Replace>-operations always include all properties of an AAA-Object, i.e also the unchanged ones. This represents a restriction of OGC's WFS-specification for the <Update> operation, which demands that at least all the changed properties are included. The reason for this restriction was the requirement that the database components do not have to track which property of an object has been changed, only the fact that an object has been changed.
- Similar to the *FeatureId* conditions, if there are multiple versions of an object, the OID *by object* must be unambiguous. This is achieved by supplementing the OID with creation date/time without separators.
- All changes carried out within aa revision case become valid at the same time. The system time (transformed to UTC) at the start of the transaction is entered into the "lifetime interval" (lebenszeitintervall) attribute of the object. The start or the end date is to be assigned on a case-by-case basis. Any data supplied for the individual features are irrelevant and will be overwritten. The latter does not apply to the initial set-up of a database if objects are transferred from a previous database (migration). In order to allow historical information to be entered, the date provided is used. If the date/time "9999-01-01T00:00:00Z" ("dummy date") is provided it is overwritten, in the same way as a revision case, with the system time. All time information is provided in UTC time (Universal Time Coordinated, Greenwich Mean Time). The time unit for entries in the lifetime interval (data type: *DateTime*) is the full second, including the mandatory identification "Z" for UTC (CCYY-MM-DDThh:mm:ssZ). The management system ensures at transfer that two versions of the same object with identical lifetime intervals cannot be created. This could occur if an object is changed, within an updating request, into several updating cases and, due to the system speed, these are processed during the same second.

The XML schema for a revision case 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 functions for updating run differently in systems with a full historical database and in systems without complete history:

Systems without complete historical database

<Insert>
The transferred features are entered as new information.

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

<Delete>:
The attribute *fid* of the filter expression in the WFS <Delete> element denotes the feature 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 deleted. 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, together with all self-referencing properties and all spatially-referenced basic forms referenced. Spatially-referenced basic forms are deleted only if they are not referenced by another 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 feature into the container.

<Replace>
The transferred features are entered, as a new version, into the container for feature versions identified by the identifier. For a unique designation of the previous version, the identifier in the filter expression (XML-attribute *fid*) of the new feature in the exchange file is supplemented by the creation date/time data of the object version to be overwritten.

This should reveal any errors occurring through updating requests that do not match the stored database. The overwritten feature remains in the receiving system as a historic version.

<Delete>
The version of the feature 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 thus historicized. 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 are described in the "Explanations about ALKIS" section.

Note: The update result currently only returns minimal information. In principle, a comparison between temporary and final identifiers plus the return of creation date/time per update case would be sensible. As this is not done in the current version, this information, when needed, for example, for encoding subsequent updates, must be requested by a subsequent primary database extract from the DHK.

5.1.2 Request for outputs

The data to be output by a data management system (utilisation data or data for managing a secondary database), in particular the scope of the information, is determined by selection and filtering criteria. A data management system must therefore be able to evaluate complex selection and filter expressions and output the data qualified by them. **Selection** is performed according to 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 always mentions only the role of the association that is marked in the application schema as navigable direction. Counter relations are not allowed in the NAS. Nevertheless, in order to simplify Filter Encoding expressions, features can be requested that are connected via counter relation with another feature. In this case the explicitly named role name of the counter relation must be used if it exists. If it does not exist the role name of the navigable direction is used and supplemented by "inversTo_".

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

The selection and filter criteria are transferred as a component of the utilisation request for the datamanaging system or logged there in the user profiles. Uniform selection and filter criteria definitions are part of the uniform product definitions of the AdV. As the formal language for defining the selection chains, the OGC *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 from OGC.

Fundamentals of Selection of Objects (Filter encoding)

To code a selection, the <wfs:Query> element from version 1.0.0 of the Open Geospatial Consortium's "Web Feature Service" (WFS) is used in the NAS. Several queries may be present in one selection, whereby each query relates to an instantiable feature type. The different queries have a complementary effect.

The current WFS specification only supports details of the concrete, instantiated 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_ActualUse" (AX_TatsaechlicheNutzung), in order to request all TN objects, instead a <wfs:Query> element must be given 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> expression consists of a predicate that is used 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 should be understood as basically having an effect on the XML instances that represent these objects.

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>

With 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> (checking for missing values)
  <ogc:PropertyIsBetween> (combination of >= and <=)

supported. The meaning of the logical operators and the comparative operators is derived from the meaning used in SQL 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 only one Bounding Box as test geometry. <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 given.

This takes place using Xpath, although we recommend limiting it 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 label</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>
  isbooked/AXpostinglocation/to/AX_postinglocation/sequentialnumber
  </ogc:PropertyName>

In cases where a relation partner in the schema is an abstract feature type (e.g. AA_ZUSO), the an instantiable feature type must be named in the Xpath expression, as in the following example.

```
    <ogc:PropertyName>

    ispartof/AX_Schwerefestpunkt/consistsof/AX_Schwere/schweresystem
    </ogc:PropertyName>
```

In this case all property paths that do not satisfy the Xpath expression are ignored. If the object to be checked in the selection is simultaneously part of a Heavy Fixed Point (Schwerefestpunkt) and another ZUSO, no properties of the other ZUSO are considered in the selection; equally, other objects which consist of the Heavy Fixed Point (Schwerefestpunkt) next to the "AX_Heavy-Object" (AX_Schwere-Objects) are not considered.

- 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>amtlicheFlaeche/@uom</ogc:PropertyName>
    <ogc:PropertyName>consistsof/AX_PointlocationAU/@srsName
(bestehtAus/AX_PunktortAU/@srsName) </ogc:PropertyName>
```

In doing so, it is assumed that the default namespace of the XML documents is "http://www.adv-online.de/namespaces/adv/gid/*version*". Otherwise, all identifiers are to be qualified by the namespace mnemonic (as illustrated 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>locationtype (stellenart) </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>2003-05-20T00:00:00Z</ogc:Literal>
    </ogc:PropertyIsGreaterThanOrEqualTo>
```

or

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

or in the case of checking for a value that is not present

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

No NULL values are included in the result set for the operator <PropertyIsNotEqualTo> of the OGC Filter Encoding. <PropertyIsNotEqualTo> and <PropertyIsEqualTo> return no complementary sets so that an additional <PropertyIsNull>request must be added.

This is, for example, the case with a request for all parcel denominators not equal to 3 on the following total portfolio: Cadastral parcel 100/1, 100/2, 100/3, 111. PropertyIsNotEqualTo> 3 </PropertyIsNotEqualTo> would deliver: cadastral parcel 100/1, 100/2 **BUT NOT** 111.

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>locationtype (stellenart) </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 m2:

```
<ogc:PropertyIsBetween>
    <ogc:PropertyName>officialarea</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>telephone</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>birthname</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>position</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"):

```
<wfs:Query typeName="AX_Gemarkung">
	<ogc:Filter>
		<ogc:PropertyIsEqualTo>
			<ogc:PropertyName>keytotal</ogc:PropertyName>
			<ogc:Literal>071234</ogc:Literal>
		</ogc:PropertyIsEqualTo>
	</ogc:Filter>
</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_Gemarkung">
	<ogc:Filter>
		<ogc:PropertyIsLike wildCard="*" singleChar="?" escape="\">
			<ogc:PropertyName> keytotal</ogc:PropertyName>
			<ogc:Literal>07*</ogc:Literal>
		</ogc:PropertyIsLike>
	</ogc:Filter>
</wfs:Query>
```

or

```
<wfs:Query typeName="AX_Gemarkung">
	<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 <wfsext: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

```
<wfsext:XlinkPropertyName traverseXlinkDepth="1">
	belongs_proportionately_to (gehoertAnteiligZu)
</wfsext: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:

```
<wfsext:XlinkPropertyPath>
        is_posted/AX_posting_location/is_part_of/AX_posting_location
</wfsext:XlinkPropertyPath>
```

Unlike with XlinkPropertyPath and PropertyName, with XlinkPropertyName the use of Xpath expressions is not allowed. Here the property of the queried feature type is to be named.

If only individual, very 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 represented in only the 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_Flurstueck">
<wfsext:XlinkPropertyPath>
            is_booked/AX_posting_location (istGebucht/AX_Buchungsstelle)
     </wfsext: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_Buchungsstelle">
        <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_Flurstueck">
        <ogc:Filter>
        <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>
                istGebucht/AX_Buchungsstelle/gml:identifier
             </ogc:PropertyName>
            <ogc:Literal>urn:adv:oid: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_Flurstueck">
        <ogc:Filter>
        <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>
                is_booked/AX_posting_location/@gml:id
             </ogc:PropertyName>
            <ogc:Literal>DEBY123412345678</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        </ogc:Filter>
    </wfs:Query>
```

The option of something like "XlinkPropertyPath" also in an inverse direction, i.e. the simultaneous selection of certain posting locations and all parcels posted to them, exists through the option of using inverse relations in filter expressions Instead of this, the selection can, of course, also 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. Access to catalogue entries for decrypting key values is frequently required.

5.1.3 Extensions to the OGC Standards

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. The following extensions are currently specified:

- Associations can be expressed by default either via the embedding of the referenced object or via an "Xlink:href reference to this. Both representations are in principle completely semantically equivalent[1]

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="urn:adv:oid:DEXXXX00000001"/>
</AX_Flurstueck>
<AX_Buchungsstelle gml:id="DEXXXX00000001">
    <zu xlink:href="urn:adv:oid: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>
                isbooked/AXpostinglocation/to/AX_postinglocation/sequentialnumber
            </ogc:PropertyName>
            <ogc:Literal>1</ogc:Literal>
        </ogc:PropertyIsEqualTo>
    </ogc:Filter>
</wfs:Query>
```

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

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

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

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

In this case, it is unclear which buildings are (should 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. This applies with both attributes and relations.

- Use of `<wfs:XlinkPropertyName>`:

  This element extends `wfs:PropertyName` with a `traverseXlinkDepth` attribute. This attribute defines to what depth `Xlink:href` references are pursued and resolved.

  A value of "1" leads to an href reference (to a local object, resolution of remote lying objects does not have to be supported) being resolved and the target object being returned in the result set. href references to this target object on the other hand are not resolved as this must be calculated at depth 2.

  A value of "*" defines that all (local) href references should be resolved. The permitted values are positive, whole numbers and "*".

  Even when an object via multiple references is resolved a number of times, it is represented only a single time in the result set.

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

  - The declaration of an Xpath expression is not permitted, it is always to be declared as an attribute of the queried feature type.
  - 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 set (feature collection).
  - In the case of NAS, the resolution of href references explicitly supports the URN identifiers of the AAA-model.
  - href references will for the time being only be resolved for locally available resources. Support for remote-Xlink-resolutions will be supplemented at a later time if required.

- Use of `<wfsext:XlinkPropertyName>`:

  In the NAS this element may replace an `ogc:PropertyName` element, however, never in an ogc:Filter Element, but only as a direct child element of a `wfs:Query` element.

It allows the selective resolution of Xlink:href references along a specific "PropertyPath", (in contrast to wfs:XlinkPropertyName, whose general expansion is supported to a specific level).

An Xpath path (Xpath-Pfadangebe) is used as value, in which an object stands at the end, at which the resolution terminated.

Example: an XlinkPropertyPath

"isbooked/AX_postinglocation/isComponentOf/AX_postingsheet" (istGebucht/AX_Buchungsstelle/istBestandteilVon/AX_Buchungsblatt)

with a query on AX_parcel (AX_Flurstueck) leads to the posting location and the posting sheet of every selected parcel being directly returned in the response set.

This element allows the specification of an attribute leafOnly. The attribute defines whether all objects along the path ("false") or only the target of the path ("true") are selected. The default here is the standard behaviour used up to GeoInfoDoc 4.0 (all objects along the path).

- Use of <wfsext:PropertyIsOfType> for checking the type of an 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.

5.1.4 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 depending on the output requirement (e.g. production of "land parcel-centred view" in ALKIS) can be achieved, whereby corresponding temporary objects are output.

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.

5.1.5 Management of secondary databases

The management of secondary databases result, case or reporting-date related, from the user-related update of the primary database. The following rules apply independently of whether the NBA processes are case or reporting-date related.

In the case of management of the secondary database without a complete historical database, i.e. only the current state of data is available in the secondary database, the following rules apply:

- The operations <wfs:Insert>, <Replace> and <wfs:Delete> are performed by the receiving system in the same logical way as for the management of primary database without a historical database.

In the case of management of the secondary database with complete historical database, i.e. in the secondary database at least the temporary and expired objects and object versions are preserved, the following rules apply:

- The operations <wfs:Insert>, <Replace> and <wfs:Delete> are performed by the receiving system in the same logical way as for the management of primary database with a complete historical database.

  Exception: In the secondary data management, object identifiers and the start of the lifetime interval of the new object version are not assigned by the system, hence these must be taken over from the "@gml:id" attribute or the lifetime interval/AA_lifetime interval/begins" (lebenszeitintervall/AA_Lebenzeitintervall/beginnt) element, which is a deviation from the rules for managing the primary database.

- In the case that an object expires (is historicized) the <wfs:Update> operator, which in other respects is not supported in the NAS, is used instead of the <wfs:Delete> operator. Only the following properties/attributes may be changed with the update:
- lifetime interval/AA_lifetime interval/ends" (lebenszeitintervall/AA_Lebenzeitintervall/endet) with the time that the last version of the object in the primary database expired. This property/attribute must be updated by every <wfs:Update> operation.
- "cause" with creation and expiry cause. For this there are two <wfs:Property< elements, each with the qualified name "cause" to be used; <wfs:Value> in the first <wfs:Property> element is the creation cause, <wfs:Value> in the second <wfs:Property> element the expiry cause. These should only be made if an expiry cause was assigned in the primary database.

Example:

```
<wfs:Update typeName="adv:AX_Flurstueck">
    <wfs:Property>
        <wfs:Name>
            adv:liftimeinterval/adv:AA_Lifetimeinterval/adv:ends
(adv:lebenszeitintervall/adv:AA_Lebenszeitintervall/adv:endet)
        </wfs:Name>
        <wfs:Value>2007-11-13T12:00:00Z</wfs:Value>
    </wfs:Property>
    <wfs:Property>
        <wfs:Name>adv:anlass</wfs:Name>
        <wfs:Value>000000</wfs:Value>
    </wfs:Property>
    <wfs:Property>
        <wfs:Name>adv:anlass</wfs:Name>
        <wfs:Value>010102</wfs:Value>
    </wfs:Property>
    <wfs:Filter>
        <ogc:FeatureId fid= "DEBY123412345678"/>
    </wfs:Filter>
</wfs:Update>
```

As an actuality check is never made in the secondary database, the management of the secondary database is specified differently from the management of the primary database, so that the fid attribute of the filter expression in <wfs:Delete>-, <Replace>- or <wfs:Update> elements is never supplemented by the creation date/time of the existing version.

This definition was selected in this way, so that as far as possible an existing Web Feature Service can be used without additional adaptation – especially in the case of history management. However it is necessary that the Web Feature Service supports the <Replace> operation of the GeoInfoDoc.

5.1.6 Locking and Unlocking Objects

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

5.1.7 Reservations

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

5.1.8 Transferring Protocol Information

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

5.1.9 Determining the Properties of a Primary Data Storage

Each software component that realizes a NAS data exchange interface must support the GetCapabilities operation.

5.2 Units to be Exchanged

The smallest units for data exchange are complete features. This also applies basically for the updating of the primary database (AAA management system). Irrespective of whether objects have qualified for output due to their own properties or due to 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 "Explanations about ALKIS").

Utilisations that do not serve the purpose of updating the primary database may, depending on user requirement or user profile, create incomplete features (missing attributes or relations) or "temporary objects" resulting from special preparation of the data 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 define as identical both the exchange interface for locations that manage a complete history and those that do not. The following general conditions, however, must be observed:

- To reduce the number of versions created, reciprocal relations in the data exchange must be presented by a single, unidirectional relation. The relations that are encoded in the data exchange are those that have been defined in the UML schema as a preferred navigation direction. Bi-directional relations in the standardised schemas are replaced, using appropriate parameterisation, by unidirectional relations.
- In order that, during data exchange, the version of an object to be overwritten 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.

5.3 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*>, also has implicit functions that enable convenient operation with the system.

The scope of the implicit data management system functionality to be implemented 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 datamanaging location.

5.3.1 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:

- With the entry of new versions, the receiving system derives the **creation date/time** from the system time. All new versions entered for an updating (or created by the <*Replace*> function) have the same creation date/time. This is usually the time when 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 features 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, an updating system cannot know whether an object that is no longer referenced by the updating can also be deleted. The **feature no longer being referenced** must be **deleted** by the database. The features 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 features that reference objects to be deleted as part of the updating process. Because counter-references are not transferred via NAS, an updating system cannot know whether an object to be deleted is referenced by further objects. As a result, it may happen 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 features that only have a right to exist if they reference other features (e.g. presentation objects). If, as part of an updating, all such references are explicitly or implicitly deleted, the data management system automatically **deletes** the corresponding **feature that lacks the necessary references**. The features 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 while updating, the database must create the geometrical and topological consistency by itself ("implicit geometry handling").
- When **deleting geometries**, break-ups from previous implicit processes are to be reversed according to the following rules. 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 are supplied, the receiving system **creates** final revision **unique identifiers**.
- At locations that do not hold a complete history, the data manager automatically **creates** 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**.

Implicit Geometry Handling

"Geometry handling" represents an implicit functionality of the database (AAA-management component) within the framework of the updating processes. In the process, new or changed geometries are linked with the existing portfolio in such a way that, with geometric identities between old and new in correlation with theme affiliation of the objects involved, redundancy free geometries are created.

Independent of the geometry handling or identity creation functions of the processing system (AAA-processingcomponent (AAA-Verarbeitungskomponente)) this functionality is always necessary if the processing system, in the course of an update, does not deliver all the objects affected by geometric operations to the AAA-Managementcomponent (AAA-Führungskomponente). (e.g. only the deleted and the new land parcels when splitting up a land parcel). For the geometry handling the following rules apply :

- The geometry handling functionality can optionally be implemented by the AAA-management systems (AAA-Führungssystemen). AAA-processing components make use, as necessary, of existing geometry handling in the AAA-management component (AAA-Führungskomponente). If geometry handling is not implemented in the AAA-Managementcomponent (AAA-Führungskomponente), the AAA-Processingcompnents must deliver complete data. Not affected by this is the commitment of the database to test the geometric consistency of the data.

- All features which are implicitly changed by geometry handling are versioned.
- The geometry handling can only be applied to features within class themes; a geometry handling for features within instance themes is not planned. 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 an identity is wanted (redundancy-free geometry) lines have to be split if necessary b) all affected features have to be delivered in the update request.
- AX_Fortfuehrungsauftrag ("update request") is supplemented with a steering parameter (geometry handling yes/no). The AAA-Managementcomponent has to evaluate this parameter and react accordingly, i.e. either turn on or off the geometry handling functionality or refuse the update request. 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 straight lines and arc/circles take part in the geometry 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.

5.3.2 Implicit functionality of a system for a secondary database

During the management of a secondary database via the NAS interface the receiving system (if required by the user) constructs and maintains the counter-references to the exchanged references.

Replace commands for which the object to be revised is not yet in the user’s database, are to be treated as *Insert* commands on receipt. (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 (I*nsert*) but from the point of view of ALKIS management system old (*Replace*), 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).

Insert commands for which the object to be entered already exists in the user’s database are to be ignored on receipt. A receiving system must be able to process object versions whose lifetime commencement lies before the interval commencement of the NBA process and was not changed during the submission period.

Due to the new form of submission, an object version may appear in various subsequent submissions with the same lifetime commencement. The receiving system must therefore recognise and ignore identical versions on receipt.

5.4 User-related Updating of Secondary Database (NBA)

This section clarifies that the following modi must be distinct according to the enumeration AX_TypeAreaTemporal (AX_Art_BereichZeitlich):

- selection of the changes to be delivered
  - "effective-date related": change-only data between last successful data submission and effective date
  - "case related" all changes between last successful data submission and effective date.
- encoding of the changes dependent on a management of history database in receiving system:
  - "without history": in the secondary database the latest state of the data is always available

– "with history": at least temporary and expired objects and object versions stored in the secondary database.

The rules for Encoding in the NAS are described in section 4.

In the combination "case related" / "with history" the data in the secondary is basically intended for use by itself for submission of requests or as source for the management of further secondary data.

5.4.1 Technical requirements

The technical requirements of the user-related updating of secondary database (NBA) are based on the procedures that exist in ALK/ATKIS- and in the ALB-system. These procedures are not identical. Further technical requirements can be summarised as follows:

Change data is to be derived from the base of revision data, which illustrates the structure of the primary database data. Change data for user-related updating of secondary database should

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

Revision-case-related means that all changes which have taken place during a previous period are listed in chronological order. 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 difference-data required to bring the user's initial state status final state desired by the user. What happened with the objects before the end state was reached cannot be reproduced. The effective-date-related change-only data represents a subset of the change data and may only be derived from it by evaluation; it comprises all newly created objects, the current versions of revised objects and details on historicized objects.

For each user, a profile is created that describes the criteria according to which the user should be provided with change data from the single database held for the NBA procedures. This profile must be created before the first data delivery.[2]

[2] The utilisation request 0040 for the first data delivery contains a profile identifier; this must be known to the system before the processing.

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 individual users; these details are also to be logged in the user profile, which is, realised by feature type AX_user group.

5.4.2 Modelling

The NBA procedure is to be offered for all feature types that a data-provider keeps in his database. The user-specific selection can be based on the whole scope of object properties; the data protection requirements have 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 transferred in several portions, the outputting system ensures that the same version of a feature 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-referenced 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 in a user-related manner.

The procedure for the user-related 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-related 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, the object identifiers must be supplemented by the creation date/time during data exchange. This requires for the following rules:

- 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 procedures with revision-related (continuous) data delivery, the relation with a creation date of 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 procedures with effective-date-related data (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, and
- 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.

5.4.2.1 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 operation**
first version of a new object	<*Insert*>
further version of an object	<*Replace*> of last transferred version (give creation date/time)
final version of an object	<*Delete*> the last transferred version (give creation date/time)

5.4.2.2 Delivery of Change-Only Data

During effective-date-related data delivery (change-only data), processing is only performed on the latest or last version of an object whose creation and/or expiry dates lie 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*> the last transferred version (indicate create date/time) with the current version of this object
final version of an object	<*Delete*> the last transferred version (give creation date/time)

5.4.3 Portioning of NBA Data, Log File

The portioning of NBA data is optional. Its use is optional and enables the recipient of NBA data to divide the overall data transfer of a large file into a number of smaller transfer steps. This was formerly very useful when transferring large data files.

The following requirements in particular are addressed: Provision of the ALKIS and ATKIS data in geometrical portions. The size of a portion should be variable but unique for one recipient (parameter for the portion size is held in the user profile). Non spatial features (NREO) and combined features (ZUSO) are selected for each portion by associations defined by selection criteria in the user profile. A portioning based on simply the amount of data is not supported.

Portion without Spatial Reference

The attribute type AX_Portionskennung.suedwestEcke (south west corner of the portion) can be assigned with a value compliant with the portioning convention or left unassigned. Depending on this, the filename either contains the above-mentioned compliant value or the filename part suedwestEcke is not assigned. The position of the non-spatially-referenced portions in the portion sequence is arbitrary.

Log File

With a portion sequence or an unportioned complete delivery, an optional log file[3] with annotations can be delivered. In the case of portioning, this is to be stated as the first portion of a portion sequence with the serial number 0. In the case of an unportioned complete delivery, the correlation between the log file and the NBA delivery is to be expressed through their file names: the name of the log file is to be the same as the name of the NBA delivery with the addition of "_p" at the end, followed by a dot with the file extension (usually ".xml", possibly also "zip", etc.). Otherwise the rules for a portion file without spatial reference (see above) apply to the log file.

5.4.3.1 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

[3] Within the context of NBA processes conducted on a huge scale, it is better to receive a log with error reports than not to know what went wrong with the NBA delivery. That is not to say that faulty NBA data is to be delivered: you then simply receive a log without data.

automatic NBA process. These are created at a specific time, taking the requirements of AX_UserGroupNBA into account. The number of NBA documents used for processing the request is left to the implementation, but a summary request protocol must be created for the one or more result files. Because of [1..n] utilization requests with cause "AX_UserRelatedDataUpdating_NBA" (0040), each portion required is delivered as an independent updating request[4].

To ensure a problem free transfer of features, the following explains the parameterisation of portioning (see below).

All portions of an utilization request contain the same application number and the same request number. The request number increases with each subsequent delivery.

For each portion, metadata must be created that define at least the geometrical and logical area of the portion.

5.4.3.2 Requirements for the sending system

Parameterising of Portioning

If portioning parameters are present in the user profile for NBA, portioning is activated but if they are absent then portioning is deactivated.

The selection and portioning is done in a two-step process:

1. The selection criteria in AX_UserGroupNBA indicate the selection of all the objects to be totally transferred in the delivery (independent of whether portioning takes place or not).

2. The separation of the selected objects into portions is driven by the portion parameters described below. Reasonable portioning parameters must be defined when creating the NBA profile.

If there is update data in a portion for the corresponding delivery, it is not necessary to create this portion. This way, the number of portions per delivery is variable, but the size of the total delivery will be clear from specifications connected with file names.

The following portion parameter can be used: **Indication of the side length** in meters. If it is applied, it is a positive integer value, counting from zero. The delivering system

[4] In this case the term "updating request" has not the same meaning as in AX_UpdatingRequest, but is of the form of a response AX_UserRelatedDataUpdating_NBA that contains the same WFS operation "transaction" as in the updating request.

automatically separates the total area, designated by the selection criteria, into corresponding squares. Hereby, the following rules apply:

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

- Collected in one portion are 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 with the corresponding spatial objects (REO).
- If one object would apply in more than one portions due to its extension, it is solely 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 only delivered in the portion in which they first appear.

Bracketing of the delivered portions

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

- in the attribute "AX_UserRelatedDataUpdating_NBA.portionCode
- in the file name of the portion.

The attribute types AX_BenutzergruppeNBA.portionierungsparameter (update user group_portion parameter and AX_NutzerbezogeneBestandsdatenaktualisierung_NBA.portionskennung (user specific data updating process_portion parameter) are optional. Without portioning they are not used and are omitted.

The portion code is created as follows:

<NBA-ProfileCode[5]>
<_>
<date of NBA creation in format jjmmtt[6]>
<_>
<sequential portion number, with leading zeros>[7]

[5] The profile code should be a clear and short description. For file names, problematic characters (e.g. space) are to be avoided as they are not accepted by the derivation, from the profile code, of this name component.

[6] Context: daily actuality of the data base. For more frequent updates, an extension with appropriately precise specifications of the time is required in the format hhmmss (i.e. without delimiters). Here the date is to be separated from the time by a capital "T" (time designator). Optionally, a capital "Z" can be used at the beginning of the time specification (time-zone designator).

[7] Encode filenames with leading zeros so that the correct sequence of the files can be more easily recognised; the sorting by filename will be incorrect without leading zeros.
xyz_070301_10of12...
xyz_070301_11of12...

<of>
<total number of portions of the delivery, no leading zeros>[8]
<_>
<pair of coordinates of the lower left corner of the corresponding portion, separated by underscore[9][10]>

Every NBA portion is marked as "consecutive no. of total no.", i.e. "2of8", for example. Leading zeros are to be encoded so that it is possible to sort by file names. The following applies here: number of leading zeros = number of digits of total number – number of digits of portion currently to be named. For example: 1000 portions in total. Portion currently to be named is the ninety-fourth, therefore two leading zeros result: 0094of1000. If in this example there was a total number of just 114 portions, the result would be 094of114, i.e. there would be just one leading zero here.

The purpose of this syntax is a) the bracketing of portions of a delivery and b) to enable the user to map the portions spatially without having to look into the NBA document.

The file names are created as follows:

- *Companyxy_20040229_124of211_3401559_5572720.xml*
- *RMR_20080229T141557Z _7of31_3401449_5573000.xml*
- *Companyxy_20040229_004of211_3401559_5572720.xml*
- *Companyxy_20040229_024of211_3401559_5572720.xml*
- *Companyxy_20040229_124of211_3401559_5572720.xml*
-

Example:

1. The selection criteria in AX_UserGroupNBA were:

xyz_070301_12of12...
xyz_070301_01of12...
xyz_070301_02of12...
xyz_070301_03of12...

[8] Total number is calculated separately for geometric and non-geometric portions. If portioning variants 001 and 002 are used together, two different total counts will be the result.

[9] Accuracy to integer meter, i.e. no decimal point. Example: "3401559_5572720"

[10] The pair of coordinates is optional. In the case of an unportioned NBA delivery it may be omitted.

```
<wfs:Query typeName="AX_PunktortTA">
    <adv: XlinkPropertyPath>isPartOf/AX_BoundaryPoint
    (istTeilVon/AX_Grenzpunkt)</adv:XlinkPropertyPath>
    <ogc:Filter>
    <ogc:BBOX>
    <ogc:PropertyName>position</ogc:PropertyName>
    <gml:Envelope srsName="urn:adv:crs:DE_DHDN_3GK3">
    <gml:posList>353100.000 5532300.000</gml:posList>
    <gml:posList>353300.000 5532500.000</gml:posList>
    </gml:Envelope>
    </ogc:BBOX>
    </ogc:Filter>
</wfs:Query>
```

2. The additional portion parameters were:
   length = 100m

3. Result:
   A maximum 4 portions are created, containing AX_PointLocationTA and the corresponding AX_BoundaryPoint

In order to enhance the overview, appropriate directory structures can be generated from profile code and date.

5.4.3.3 Requirements for the receiving system - Processing of the delivery

Based on the profile code ("3of8") check whether all potions of one delivery have been transferred. The addressed system must know which portions comprise the total delivery. Only after the complete transfer of an NBA delivery can it be receipted, on request, and the transfer of the next delivery started.

5.4.4 Receipting NBA deliveries

After the transfer of an NBA delivery, a receipt can be sent to the delivering location using an NBA receipt-request AX_NBAReceipt (NBA-Quittierungs-Auftrags AX_NBAQuittierung), so long as this is state-specifically required and requested in the NBA delivery. The processing of the NBA receipt has to conform to state-specific rules.

6 AdV Definitions for Metadata

Metadata on georesources contains descriptive information and thus makes statements about the characteristics of datasets, dataset series, geographic data services and applications, and their structure and the correlations in their contents. Metadata is publicly accessible data and makes it possible to find and access specific geographic data. By its informative character, it makes it possible to avoid redundant data collection, detect existing gaps in the databases, standardise data and terms, manage quality for the datasets, compare alternative databases and create transparency in the data market.

6.1 Metadata for the European Geographic Data Infrastructure INSPIRE

With Directive 2007/2/EC, the European Commission resolved for a European geographic data infrastructure to be established for Community policies ("INSPIRE"). A crucial component of this is locating georesources, i.e. data and services by means of metadata. For the establishment of a European geographic data infrastructure, COMMISSION REGULATION (EC) No. 1205/2008 of 3 December 2008, implementing Directive 2007/2/EC of the European Parliament and of the Council as regards metadata, prescribes a structure and a defined minimum quantity of metadata elements, which have also been taken fully into account in the AdV metadata profile. Furthermore, metadata is contained, related to datasets, in the data specifications of INSPIRE. These metadata are also contained in the AdV metadata profile.

6.2 The Metadata Standard ISO 19115

The standard *ISO 19115 Geographic Information: Metadata* provides a very wide spectrum of textual descriptions of geospatial data.. The standard contains more than 400 metadata elements that serve to describe geodata and are defined as either mandatory, conditional or optional. To achieve ISO conformity, in particular all the mandatory elements must be filledwith due regard to conditional correlations and cardinalities. User groups can define any subsets (profiles) for their particular requirements in compliance with ISO conformity. At the same time, the ISO schema can also be expanded with additional individual elements (extensions).In its present form, however, ISO 19115 does not take into account metadata about services. Contained in ISO 19119 there is a chapter listing and describing the special metadata for the documentation of services that goes beyond the scope of ISO 19115.

6.3 AdV Metadata Profile

The GeoInfoDoc references a metadata profile of ISO 19115 in which all the elements used for the AAA application schema are listed. This covers both object-related metadata (e.g. quality information of points) in the AAA application schema and database-related metadata. The AdV metadata profile essentially contains a table along the lines of ISO 19115 with German translations of the terms and definitions and explanations for their use in the context of the AdV.

The AdV metadata profile is published as a standalone document. For this, on the basis of ISO 19115 and 19119, the elements relevant to the AdV were extracted, translated and elucidated and form the AdV metadata profile. The translation from English of the names and descriptions generally draws on the translation by a team from GDI-DE.

Furthermore, the stipulations and guidelines given by INSPIRE were taken into consideration, i.e. a metadata set according to the AdV metadata profile is simultaneously in conformity with INSPIRE.

The AdV metadata catalogue was hitherto part of the GeoInfoDoc up to and including version 6.0.1. Due to the universal character of the metadata (it applies not only to AAA data and AAA products), in the future the AdV metadata profile will hence be maintained and further developed independently and detached from the GeoInfoDoc version logic, as part of the series of AdV profiles (such as the AdV-WMS profile). All the elements necessary for the AAA application schema are contained in the AdV metadata profile. The metadata elements are used in the GeoInfoDoc in the following contexts:

- Object-related metadata, usually modelled as an integral model component within attribute types or as data types within the object types
- Product-related metadata that can be delivered as part of the output
- Metadata of the AdV in the metadata information system (AdV-MIS) for the standardised description of the data and services of the AdV. This is the total of all the metadata elements required by AdV including the INSPIRE metadata.

Product-related metadata is divided into static and dynamic metadata. Static metadata is generally assigned once and changes rather infrequently (e.g. contact to a service centre). Here the valid AdV metadata profile is referred to, i.e. this metadata is not an integral model component but rather a component of the AdV-MIS. Nonetheless, it can be requested via a utilisation request and released to users.

There are three options for managing product-related static metadata:

- It can be kept in the same data storage as the primary database data. Advantage: Data storage is autonomous, i.e. it does not have to wait for an external metadata server when the product is being created. Need for action: Data storage has to be further developed into a metadata information system (MIS) if an universal search is required for this. In addition, the data storage needs to be expanded to include appropriate metadata aspects.

- It can be kept in a data storage external to the AAA primary database (e.g. on a metadata server). Advantage: The product-related static metadata can be searched for universally. Disadvantage: Two data storage systems are made to depend on one another (waiting times for metadata transfer).

- Compromise: Metadata in external data storage and in the same data storage as the primary database data in the sense of synchronised databases (although the AdV does not stipulate which database is the primary database and which is the secondary one).

Which variant is used depends on the implementation in each case.

Product-related metadata is composed of that which is in the data model regaded as static information (structured according to ISO 19139, including query, transport and maintenance via CSW) and those data which are dynamic, which means that they are calculated for a specific selected data set and not applicable for the whole data set (e.g. the average psotional accuracy of the selected boundary points). The extent and content of the metadata elements released to the users are determined product-specifically by the expert working groups of the AdV.

Dynamic metadata: Is generated during the run time to get from non-persistent information such as "time of executing a utilisation request" a specific metadata information for a utilisation response.

These things must be defined from a technical point of view and integrated, as necessary, in the AAA application schema. The set-up and updating of metadata databases is therefore not yet part of the modelling or definitions.

7 Registry

According to ISO 19135, a registry is an information system on which a register is maintained, while a register is a classification of so-called identifiers for resources and their descriptions. In this context, a resource is a fact that can be described unambiguously to distinguish it from other facts. An identifier makes it possible to reference resources from the registry unambiguously.

Registries take on a central role in geodata infrastructures as they enable the administration, location and use of the geoinformation resources available in the infrastructure. These include, for example, data products, services, application schemas (in UML/XML), coordinate reference systems, units of measure, code lists, data and feature types, service types and also character sets and symbols. The resources themselves as well as the important distinguishing characteristics that are especially important for the administration, location and usage shall be made available via a registry.

7.1 Registry for Coordinate Reference Systems and Units of Measure

Due to the technical necessity in the area of the AdV, a web-based register for coordinate reference systems is currently being established on behalf of the working group Spatial Reference (AK RB) of the AdV in order to supply reliable CRS descriptions from an official source. Implementation is being carried out by the project groups AFIS (AK RB) and GDI Standards (AK IK) of the AdV. There is ongoing communication with the GDI-DE pilot project Registry. Implementation is based on the ISO standards

- ISO 19111 Spatial referencing by coordinates
- ISO 19127 Geodetic codes and parameters
- ISO 19135 Procedures for item registration.

As resources for implementation, resources were selected that are used in almost every NAS file:

- Coordinate reference systems (abbreviation: CRS) with the associated components such as coordinate systems, date, coordinate operations, etc.
- Units of measure (abbreviation: UoM) that are also used by the coordinate reference systems

For appropriately authorised bodies, a web interface provides a user interface as an information system with a search function and an interface for maintaining the AdV CRS register.

The AdV CRS register currently stipulates the following participants:

Role	Role fulfilled by
Register owner (Owner of the CRS registry)	AdV
Operational running of the registry	BKG
Register manager (Body technically responsible for the actual maintenance)	AAA Review Committee of the AdV
Control body (Technically responsible body that decides on new entries)	PG AFIS of the working group Spatial Reference of the AdV
Submitting organisations (Body permitted to submit change requests)	AAA Review Committee / BKG / Federal States
User	Anyone, publicly available

The AdV registry for CRS and units of measure is available in prototype form and, in cooperation with the GDI-DE (SDI Germany) operating unit, it is to be further developed into a central component of the geodata infrastructure in Germany. Consequently, the roles listed could still change.

A registry service is not to be implemented for the time being. According to the current policy position of the AdV, it is currently not yet planned to replace the CRS list of this document with the CRS registry. In view of this, the GeoInfoDoc continues to have a normative character with regard to the CRS in Germany.

7.2 Registry for Code Lists

Characteristics of geo-objects that are frequently used or are to be described according to a predefined concept are often depicted through codes in the AAA data model. As a rule, the number of possible values is specified in a code list. The code list is usually published together with the data model. Data entry staff need to be familiar with this code list, the meaning of the individual codes and the state of maintenance and quality so as to be able to select the appropriate value for data entry.

In some cases, however, the codes available in a code list are not adequate. In the AdV model sometimes only the collective view is covered but not the Federal State specific needs. In this case, the AdV allows for certain code lists to be expanded, also at a Federal State specific level. In order to avoid the allocation of the same codes, thus resulting in inconsistent datasets, it is necessary to organise the maintenance of the Federal State specific code lists (e.g. model types). For this a registry is a suitable instrument that makes it possible to organise the maintenance of national and Federal State specific code lists as well as other code lists.

The use of a register for code lists supports

- the responsible bodies with the maintenance of the code lists and the codes
- the data entry staff in selecting suitable codes and compiling harmonised geodata sets
- the users in interpreting the codes in a dataset.

A registry can contain various resources. If the resources are the code lists of the AAA application scheme, the term “code-list and enumeration registry” is used. This term is abbreviated as CER. First the ISO and AdV definitions and regulations on the topic are presented, followed by the actual CER concept.

7.2.1 Definitions and Regulations Regarding Code Lists and Enumerations

Definition of code list:

Range of values in which a code is allocated for each permissible value (ISO 19136). Differentiation between the term "enumeration" and "code list" according to ISO 19103 (211n2763_Review_Final_CD_19103):

An enumeration is an exhaustive collection of permissible values assigned the stereotype <<enumeration>>. Enumerations cannot be expanded.

A code list is a non-exhaustive collection of permissible values assigned the stereotype <<codeList>>. Code lists can be expanded.

ISO 19103 distinguishes between two types of code lists: those with the Tagged Value "codeList" (these are managed by a single responsible body) and those without this Tagged Value (these can be expanded at will by any user of the related application schema). Further information is found in chapter 3.3.8 "Expandable 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 generally defined as abstract classes and assigned the stereotype <<CodeList>> in the basic schema. In some cases the code lists in the basic schema have values specified (e.g with AA_initiationtype (AA_Anlassart)). Expansions and changes to these lists do not result in a new version of the data exchange specification. Thus, rather than being published in the output schema, they are managed in a dictionary format in an "external" XML file. They are managed and maintained at a central location with the option of online access.

The demand for a concept for expanding these code lists grows at the same rate as implementations of the AAA model. As soon as, for example, Federal State XYZ has set up a Federal State specific output along the lines of the AdV guidelines for technical data connection, the code in code list AX_initiationtype_utilisationrequest (AX_Anlassart_Benutzungsauftrag) needed to initiate the creation of exactly this output is found to be missing.

All code lists of the AAA technical schema use four or six-digit whole number codes (odd integers due to leading zero). The code lists with the following number of digits can be considered for a Federal State specific expansion:

Code list	**Number of digits**
AA_Anlassart (AA_initiationtype)	6
AA_Anlassart_Benutzungsauftrag (AX_initiationtype_usageorder)	4
AA_WeitereModellart	unlimited

The Federal State specific expansion of AAA technical schema code lists (especially the codes) is introduced with the two-character Federal State abbreviation (cf. chapter "Identifiers, Links"). The abbreviation "BKG" represents the Federal Office of Cartography and Geodesy (BKG).

As additional characters, digits {0-90} and letters {A-Z, a-z, without umlauts} are allowed. The string is case sensitive.

In order to simplify the implementation, the position count of the Federal State specific code, including prefix, should be identical to that of the AdV code. In the future, there may be a need for larger position counts for technical information systems.

Examples:

- A Federal State specific four character usage initiation for example would be "RP10" or "RPA6".
- A four character BKG usage initiation (the only occurrence of a three character initial) would be "BKG7" or "BKGa".

This simplifies the central registration ("Registry") of the expandable code lists (every Federal State and the BKG operates in a single namespace). If the registry is not required within the framework of GDI-DE, it can even be completely omitted.

An additional requirement for managing a CER arises from section 3.5 Quality Data and Metadata:

"In the role information a code list pointer is required that, according to ISO/TS 19139 8.5.5, must be a URL. In the example a URL of a Code-List-Dictionary in the OGC-Schema repository is given. Alternatively, this can also be – as with schema pointers – another valid pointer to a Code-List-Dictionary."

This is illustrated in greater detail with an example:

```
"<gmd:role>
        <gmd:CI_RoleCode
                codeList="http://schemas.opengis.net/iso/19139/20070417/resources/Codelist/gmxCod
                elists.xml#CI_RoleCode" codeListValue="processor">processor</CI_RoleCode>
</gmd:role>"
```

7.2.2 Construction of a Temporary Registry

For the publication of the code lists and enumerations of the GeoInfoDoc, the AdV is initially providing a registry of a temporary nature. The registry is limited to the scope of the AAA application schema. It does not allow any expansions or modifications. Following the availability of the GDI-DE registry, which is currently being implemented, the contents are to be migrated to the GDI-DE registry and the maintenance of the code lists is to be carried out in this context. This applies both to the code lists and enumerations from the AAA application schema and to technical models based on this, such as TFIS or expansions in the Federal States. The procedure here is as follows:

- Code lists and enumerations from the AAA application schema are maintained in the UML model of the AAA Review Committee.
- The code lists and enumerations of the current reference version 6.0.1 are published, along with all modifications following the adoption of version 7.0.0.

- The temporary registry consists of files that are made available by a web server.
- The following files are made available:
  - Registry: list of all code lists and enumerations in the registry
    - HTML: for display in a web browser
    - ATOM: for processing with software or for display in feed readers
  - Code list / enumeration: list of all value types
    - HTML: for display in a web browser
    - ATOM: for processing with software or for display in feed readers
    - GML: for processing with software
    - SKOS/RDF: for processing with software
  - Value type: information on the value type
    - HTML: for display in a web browser
    - ATOM: for processing with software or for display in feed readers
    - GML: for processing with software
    - SKOS/RDF: for processing with software
- The entries contain both the information on the registered object and the registry metadata. The core registry schema from the current revision of ISO 19135-1 is used, extended with the data model for the code list register in the GDI-DE registry and with necessary additional information (model types, inclusion in core data inventory, from/to GeoInfoDoc version).
- The files are derived by means of a ShapeChange extension.
- The registry is managed locally and only the external representations of the code lists, enumerations and value types that are generated by the registry in the form of files shall be published on the web. To this end, the files are copied onto an appropriate web server. To support HTTP content negotiation, an Apache configuration is created.
- The URIs under which the registry is accessible and with which the entries in the registry are identified are only of a temporary nature prior to migration to the GDI-DE registry. Following a migration to the GDI-DE registry, enquiries to the temporary URIs should be redirected to the persistent GDI-DE registry URIs via HTTPRedirect for at least three years.

7.3 XML Schema Register

In the case of OGC and INSPIRE, XML schemas are provided in simple file-based repositories via web servers. This simple provision is adequate for the purposes of the AdV. Here no register in line with ISO 19135 is required; instead, a simple solution is sufficient. Since GDI-DE has already set up an appropriate register, all the NAS files have been placed there. See http://repository.gdi-de.org/schemas/adv.

7.4 Enterprise Architect Subversion-Server

The AAA application schema has stopped being managed locally on the computers of the AAA Review Committee members since the changeover to the modelling tool Enterprise Architect; instead it is managed centrally in a UML schema repository on a server. Thus it counts as a further centrally managed register of the AdV. The advantage lies in the clear reference version and the avoidance of version conflicts thanks to an explicit version concept and restricted access to using the register. Furthermore, all changes by all parties are documented.

In addition, besides publication in AdV-online, there is the option of setting up reading access for additional persons (e.g. AAA representatives of the Federal States).

The ISO data models are also incorporated via an EU Subversion server that is hosted at the Joint Research Center (JRC).

8 Coordinate Reference System and Measurement Units

8.1 Coordinate Reference System for AFIS-ALKIS-ATKIS

8.1.1 System Used

In AFIS-ALKIS-ATKIS the associated coordinate reference system (CRS), for each geometry, can be given or stored. The descriptions (short) used are defined in this section.

The **short description** is assembled from the following information:

[*Land*] _[*geodätisches Datum*] _[*Koordinatensystem*] _[*Submerkmale des Koordinatenreferenzsystems (z. B. Lagestatus)*]

The short descriptions to be used are defined in the tables below.

8.1.2 Coordinate Reference System for 2D Locational Information

Preliminary remarks:

1. The coordinate values of the CRS are given in the following order:
   - for Gauß-Krüger projection: easting, northing
   - for UTM projection East, North, and
   - for Lambertscher Kegelabbildung: East, North, and
   - for geographical coordinates latitude, longitude.

2. The place holders <sn> and <zn> are to be replaced by the number of stripes (for Gauß-Krüger) or the Zone (for UTM, without letter code) respectively. Therefore each stripe or zone has its own CRS defined. In the register the parameter "false easting" is to be allocated the value 500000 m and "zone number" the value of the respective Zone or stripe

   Example:

DE_DHDN_3GK2 (easting, northing):	581996. 560	5616134. 450
ETRS89_UTM32 (East, North):	369949. 671	5615301. 383

3. In order to simplify the evaluation (e.g. of coordinate lists) during presentation of the standard output, the coordinate information still contains the stripe or zone codes e.g.:

GK coordinates (easting, northing):	2581996. 560	5616134. 450
UTM coordinates (East, North):	32369949. 671	5615301. 383

Main Group	Sub group	Country	Short description
DHDN, Lambert Conformal Conical Projection		DE	DE_DHDN _Lam
DHDN, ellipsoidal (geodetic) coordinates		DE	DE_DHDN_Lat-Lon
DHDN, Gauß-Krüger-3-degree stripes		DE	DE_DHDN_3GK<sn>
	old location point field (Reichs triangle network (Reichsdreiecks-netz))	DE	DE_DHDN_3GK<sn>_RDN
		BY	DE_DHDN_3GK<sn>_BY120
		BE	DE_DHDN_3GK<sn>_BE200
		HH	DE_DHDN_3GK<sn>_HH100
		HE	DE_DHDN_3GK<sn>_HE120
		NI	DE_DHDN_3GK<sn>_NI200
		NW	DE_DHDN_3GK<sn>_NW101
		RP	DE_DHDN_3GK<sn>_RP101

Main Group	Sub group	Country	Short description
		SH	DE_DHDN_3GK<sn>_SH200
		TH	DE_DHDN_3GK<sn>_TH200
		SL	DE_DHDN_3GK<sn>_SL159
	completely renewed, countrywide system		
		BW	DE_DHDN_3GK<sn>_BW100
		HB	DE_DHDN_3GK<sn>_HB100
		NI	DE_DHDN_3GK<sn>_NI000
		NI, ST	DE_DHDN_3GK<sn>_NI100
		NW	DE_DHDN_3GK<sn>_NW177
		RP	DE_DHDN_3GK<sn>_RP180
		HE	DE_DHDN_3GK<sn>_HE100
		SL	DE_DHDN_3GK<sn>_SL197
	partially renewed system		
		BY	DE_DHDN_3GK<sn>_BY110
		HB	DE_DHDN_3GK<sn>_HB110
		HE	DE_DHDN_3GK<sn>_HE110
		SH	DE_DHDN_3GK<sn>_SH210
		NI	DE_DHDN_3GK<sn>_NI210
		NW	DE_DHDN_3GK<sn>_NW119
		NW	DE_DHDN_3GK<sn>_NW131
		NW	DE_DHDN_3GK<sn>_NW133
		NW	DE_DHDN_3GK<sn>_NW158
		NW	DE_DHDN_3GK<sn>_NW163
		NW	DE_DHDN_3GK<sn>_NW166
		NW	DE_DHDN_3GK<sn>_NW173
		NW	DE_DHDN_3GK<sn>_NW174
		NW	DE_DHDN_3GK<sn>_NW175

Main Group	Sub group	Country	Short description
		NW	DE_DHDN_3GK<sn>_NW176
System 40/83, GK-3-degree		BB, ST, MV, SN,TH,	DE_40-83_3GK<sn>
System 42/63, GK-6-degree		BB, ST, MV, SN,TH, Eastern Europe	DE_42-63_3GK<sn>
System 42/83, GK-6-degree		BB, ST, MV, SN,TH, Eastern Europe	DE_42-83_6GK<sn>
System 42/83, GK-3-degree		BB, ST, MV, SN,TH, Eastern Europe	DE_42-83_3GK<sn>
System 42/86, GK-3-degree		BB, ST, MV, SN,TH, Easterm Europe	DE_42-86_3GK<sn>
System 42/83, ellipsoidal (geodetic) coordinates		BB, ST, MV, SN,TH, Eastern Europe	DE_42-83_Lat-Lon
RD/83, GK-3-degree		SN, BB, ST, MV	DE_RD-83_3GK<sn>
RD/83, ellipsoidal (geodetic) coordinates		SN, BB, ST, MV	DE_RD-83_Lat-Lon
PD/83, GK-3-degree		TH	DE_PD-83_3GK<sn>
PD/83, ellipsoidal (geodetic) coordinates		TH	DE_PD-83_Lat-Lon
Cadastre System of the Prussian National Topographic Survey			
	System Baden	BW	DE_Soldner-Baden
	System Württemberg	BW	DE_Soldner-Wuerttemberg
	System Berlin	BE	DE_Soldner-Berlin
	System 18 Müggelberg	BE	DE_Soldner-Mueggelberg
	System 17 Greifswald	MV	DE_Soldner-Greifswald
	System 24 Ostenfeld	SH	DE_Soldner-Ostenfeld
	System 25 Rathkrügen	SH	DE_Soldner-Rathkruegen
	System 26 Bungsberg	MV, SH	DE_Soldner-Bungsberg
	Soldner Grossenhain1	SN	DE_Soldner-Grossenhain1
	Soldner Grossenhain2	SN	DE_Soldner-Grossenhain2
	Soldner	SN	DE_Soldner-Grossenhain3

Main Group	Sub group	Country	Short description
	Grossenhain3		
	Soldner Leipzig	SN	DE_Soldner-Leipzig
	Soldner Torgau	SN	DE_Soldner-Torgau
Mecklenburgisches Coordinates System 1912		MV	DE_Mecklenburg_1912
System Hamburg old		HH	DE_Hamburg_220
System Hamburg new		HH	DE_Hamburg_210
System ED50/UTM		Europe	ED50_UTM<zn>
System ED50, ellipsoidal (geodetic) coordinates		Europe	ED50_Lat-Lon
System ED87/UTM		Europe	ED87_UTM<zn>
ETRS89/UTM		Europe	ETRS89_UTM<zn>
ETRS89/GK-3-degree		Europe	ETRS89_3GK<sn>
ETRS89, ellipsoidal (geodetic) coordinates		Europe	ETRS89_Lat-Lon
ETRS89, Lambert Conformal Conical Projection		Europe	ETRS89_LCC
ETRS89, Lambert Conformal Conical Projection		DE	ETRS89_Lam

8.1.3 Coordinate Reference System for 3D Locational Information

Main Group	Sub group	Country	Short description
DHDN, ellipsoidal (geodetic) coordinates incl. ellipsoidal elevations		DE	DE_DHDN_Lat-Lon-h
System 42/83, ellipsoidal (geodetic) coordinates incl. ellipsoidal elevations		SN	DE_42-83_Lat-Lon-h
ETRS89, ellipsoidal (geodetic) coordinates incl. ellipsoidal elevations		Europe	ETRS89_Lat-Lon-h
ETRS89/UTM incl. ellipsoidal elevation		Europe	ETRS89_UTM<zn>-h
ETRS89/GK-3-degree incl. ellipsoidal elevation		Europe	ETRS89_3GK<sn>-h
ETRS89, spatial Cartesian coordinates		Europe	ETRS89_X-Y-Z
WGS84, spatial Cartesian coordinates		World	WGS84_X-Y-Z
WGS84, ellipsoidal (geodetic) coordinates incl. ellipsoidal elevations		World	WGS84_Lat-Lon-h
WGS84/UTM incl. ellipsoidal elevation		World	WGS84_UTM<zn>-h

8.1.4 Coordinate Reference System for Elevations

Main Group	Sub group	Country	Short description
Old systems or reference gauges			
	Old Hamburg Datum (Alt-Hamburger Null), main tide gauge in Hamburg 1841	HH	DE_ALT_HH_1841
	New Hamburg Datum (Neu-Hamburger Null), main tide gauge in Hamburg 1872	HH, SH	DE_NEU_HH_1872
	Mean sea level of the Baltic Sea 1840 near Swinemünde	MV	DE_MWO_1840
	Mean sea level of the Baltic Sea 1875 near Swinemünde	SN	DE_MWO_1875
	Datum at Neufahrwasser near Danzig	Baltic Sea area	DE_ALT_NWD
	Cuxhaven Datum (Cuxhavener Null) at the main tide gauge	NI	DE_ALT_CUX
	Harburg Tide Gauge Datum (Harburger Flutmessernullpunkt) until 1937	HH	DE_ALT_FMN
	Helgoland Datum (Helgoländer Null) (H.N.)	SH	DE_ALT_HELG
	Amsterdams Peil (AP) 1818	Europe	EU_ALT_AP
	Normaal Amsterdams Peil (NAP) from 1891	Europe	EU_NAP
Old or provisional system, NN elevation via NHP 1879			
	Old system, NN elevation via NHP 1879, without levelling reduction	DE	DE_ALT_NN
	Old system in Baden,	BW	DE_ALT_NN_BW010

Main Group	Sub group	Country	Short description
	NN elevation via NHP 1879, without levelling reduction		
	Old system in Württemberg, NN elevation via NHP 1879, without levelling reduction	BW	DE_ALT_NN_BW020
	Provisional system, NN elevation via NHP 1879, normal orthometric elevation	BY	DE_VORL_NOH_BY901
DHHN12 (previously: "New System"), NN elevations via NHP 1912, network parts I to VIII			
	DHHN12, Normal orthometric elevation	DE	DE_DHHN12_NOH
	completely renewed, countrywide systems		
	DHHN12, Horizon 55, normal orthometric elevation	NI	DE_DHHN12_NI120
	DHHN12, Horizon 71, normal orthometric elevation	BW	DE_DHHN12_BW130
	DHHN12, System 68-74, normal orthometric elevation	RP	DE_DHHN12_RP120
NKN	North Sea Coast Levelling (NKN) I (1928-1931), normal orthometric elevation	HB, HH, NI und SH	DE_ NKN-I_NOH
	North Sea Coast Levelling (NKN) II (1949-1955), normal orthometric elevation	HB, HH, NI and SH	DE_ NKN-II_NOH
	DHHN12, Northwest Europe Flatland Levelling (NWELL) (1949-1956), normal orthometric elevation	NI	DE_DHHN12_NOH_NWELL
	DHHN12, Northwest Europe Flatland Levelling (NWELL) (1949-1956), Geopotential altitude	NI	DE_DHHN12_CP_NWELL
OKN	Provisional system, Baltic Sea Coast	DE	DE_OKN-I_NOH

Main Group	Sub group	Country	Short description
	Levelling (OKN) I (1896-1901), normal orthometric elevation		
Levelling network 1960			
	Levelling network 1960, Normal orthometric elevation	DE	DE_NIV60_NOH
	Levelling network 1960, Horizon 74, Normal orthometric elevation	HB, NI	DE_NIV60_NOH_NI130
	Levelling network 1960, Horizon 77, Normal orthometric elevation	HB, NI	DE_NIV60_NOH_HB131
	Levelling network 1960, Geopotential altitude	DE	DE_NIV60_CP
DHHN85			
	DHHN85, Normal orthometric elevation, Wallenhorst Datum Point, Underground Marker I	DE	DE_DHHN85_NOH
	DHHN85, Geopotential altitude, Wallenhorst Church Datum Point, benchmark	DE	DE_DHHN85_CP
DHHN92			
	DHHN92, Normal elevation	DE	DE_DHHN92_NH
	DHHN92, Geopotential altitude	DE	DE_DHHN92_CP
DHHN2016	DHHN2016, Normal elevation	DE	DE_DHHN2016_NH
	DHHN2016, Geopotential altitude	DE	DE_DHHN2016_CP
	DHHN2016, Normal orthometric elevation	DE	DE_DHHN2016_NOH
SNN56			
	SNN56, Normal elevation	BB, ST, MV, SN,TH	DE_SNN56_NH
	SNN56, Normal orthometric elevation	BB, ST, MV, SN,TH	DE_SNN56_NOH
SNN76			

Main Group	Sub group	Country	Short description
	SNN76, Normal elevation	BB, ST, MV, SN,TH	DE_SNN76_NH
	SNN76, Normal orthometric elevation	ST	DE_SNN76_NOH
	SNN76, Geopotential altitude	BB, ST, MV, SN,TH	DE_SNN76_CP
DHDN, Ellipsoidal elevation		DE	DE_DHDN_h
	Heitz-Geoid	NI	DE_Bessel_h_NI700
	Lelgemann-Geoid	NI	DE_Bessel_h_NI710
United European Levelling Network (UELN) 73/86			
	UELN73/86, Normal elevation	Europe	UELN73-86_NH
	UELN73/86, Geopotential altitude	Europe	UELN73-86_CP
European Vertical Reference System (EVRS) 2000, United European Levelling Network (UELN) 95/98			
	UELN95/98 (EVRF2000), Normal elevation	Europe	EVRF2000_NH
	UELN95/98 (EVRF2000), Geopotential altitude	Europe	EVRF2000_ CP
European Vertical Reference System (EVRS) 2007, United European Levelling Network			
	EVRF2007, Normal elevation	Europe	EVRF2007_NH
	EVRF2007, Geopotential altitude	Europe	EVRF2007_CP
WGS84, Ellipsoidal elevation		World	WGS84_h
ETRS89, Ellipsoidal elevation		Europe	ETRS89_h
System 42/83, Ellipsoidal elevation		SN	DE_42-83_h
Height anomaly (quasigeoid elevation)			
	EGG97	Europe	EGG97_QGH
	GCG2005	DE	DE_AdV_GCG2005_QGH

Main Group	Sub group	Country	Short description
	GCG2011	DE	DE_AdV_GCG2011_QGH
	GCG2016	DE	DE_AdV_GCG2016_QGH

8.1.5 Combinations of Coordinate Reference System for Location and Elevation

Combination of location and elevation systems (Compound coordinate reference system, CCRS) are always cited by concatenating the codes of the components using an "*", e.g.:

DE_DHDN_3GK2_RDN*DE_DHHN92_NH

Nevertheless, for features of feature type "AX_PointLocation" combined coordinate reference systems are **not** allowed in AFIS-ALKIS-ATKIS.

8.1.6 Designation of the Coordinate Reference System in the NAS

The designation of the CRS in the NAS (GML) has the data type "anyURI". In this way both URL and URN designations are allowed. The URL variant assumes an explicit XML description of the CRS used in a file. As this is not yet available, the CRS will be referenced for the time being via a URN as follows:

srsName="urn:adv:crs:*short description*"

As soon as the corresponding description of the CRS is available the alternative URL will be used so that the CRS will be referenced as follows:

srsName="http://www.adv-online.de/crs/crs.xml#*short description*".

In the NAS, the coordinate designations for Gauß-Krüger and UTM coordinates do not include stripe or zone designations, e.g.

Gauß-Krüger coordinates (easting, northing):	581996.560 5616134.450
UTM coordinates (East, North):	369949.671 5615301.383

In the NAS these look like this:

```
...
<gml:Point srsName="urn:adv:crs:DE_DHDN_3GK2_NW177">
            <gml:coordinates>581996.560 5616134.450</gml:coordinates>
</gml:Point>
<gml:Point srsName="http://www.adv-online.de/crs/crs.xml#DE_DHDN_3GK2_NW177">
            <gml:coordinates>581996.560 5616134.450</gml:coordinates>
</gml:Point>
...
```

or

```
...
<gml:Point srsName="urn:adv:crs:ETRS89_UTM32">
            <gml:coordinates>369949.671 5615301.383</gml:coordinates>
</gml:Point>
<gml:Point srsName="http://www.adv-online.de/crs/crs.xml# ETRS89_UTM32">
            <gml:coordinates>369949.671 5615301.383</gml:coordinates>
</gml:Point>
...
```

8.2 Units of Measure for AFIS-ALKIS-ATKIS

8.2.1 System Used

In AFIS-ALKIS-ATKIS a measurement unit must be specified for each quantitative value. The short descriptions to be used are defined in this document.

If, in future, a corresponding register of measurement units is administered by ISO, the Open Geospatial Consortium (OGC) or any other organisation, a changeover to the entries defined there is envisaged.

8.2.2 Short Descriptions

Measurement	Short description
Metre	m
Millimetre	mm
Kilometre	km
Square metre	m2
Cubic metre	m3
Degree, decimal (old degree)	degree
Gon, decimal	gon
Radians	rad
m/s^2	ms-2
m^2/s^2	m2s-2

8.2.3 Designation Unit of Measure in the NAS

The designation *Unit of Measure* in the NAS (GML) has the data type "anyURI". In this way both URL and URN designations are allowed. The URL variant assumes an explicit XML description of the unit of measure used in a GML dictionary. As this is not yet available, the CRS will be referenced for the time being via a URN as follows:

uom="urn:adv:uom:*short description*"

As soon as the corresponding description of the unit of measure is available the alternative URL will be used so that the CRS will be referenced as follows:

uom="http://www.adv-online.de/uom/uom.xml#*short description*".

9 Quality assurance

9.1 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 conformance 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.

9.2 Quality assurance model

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

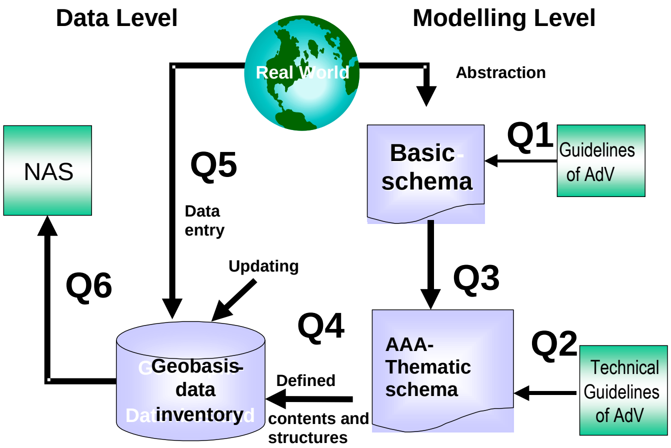


Figure 40: The quality assurance model of the AFIS-ALKIS-ATKIS project

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 prin-ciples	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). When testing exchange data against the NAS-schema, a distinction is drawn between testing for a well-formed XML file and testing for validity of the XML file (test tool for both e.g Xerces).

9.3 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:

:

Documents about Quality Management
Quality assurance of the common AAA-technical schema against the technical stipulations of the AdV (Q2)
Quality assurance of the common AAA-technical schema against the AAA-basic schema (Q3)
Appendix to Q3
Quality assurance of the exchange data against NAS (Q6)

10 Glossary, Abbreviations

10.1 Technical terms and their English translations

Technical term (German)	Explanation	Technical term (English)
AdV Standard	The AdV creates rules for the development of *Procedures* and *Program systems* and for the creation of *Products*. Through the commitment of the member authorities to their compliance, AdV rules, used to specify state-independent core data, data exchange interfaces and standard products, are elevated to AdV standards.	**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 acquisistion sources, primary database data and also their digital and analogue extracts from AFIS, ALKIS and ATKIS and also deivery 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 is administered as an attribute of AA-object with the object identifier and the lifetime interval.	**cause (for a change)**
Application schema	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**
Auszüge	Extracts are data inventories selected according to content, area and/or time period (e.g. updating data inventories), which are delivered to the user as data structured by object or image or as analogue extracts.	**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**
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**
Collection data	The collected data represents the basis for revising the official geoinformation. It is formed by collection from source data collated through reconnaissance methods in the real world or recorded from cartographic representations and other documents.	**Collected data**
Datenmodell	A data model describes the fundamental properties that simplify a unified representation for all appearances of a certain (thematically oriented) 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**
Detailierungs-grad	The *Level of detail* defines the geometric and thematic resolution of 3D objects. In ALKIS®, the differential between levels of detail is dependent on the geometry or texturing.	**Level of detail**
Differenzdaten	Change-only data is change data relating to the effective date required to bring the initial status of the primary database data up to the required final status (effective date) for the user. It consists of all newly created objects, the current versions of revised objects and details on historicized objects. The differential updates represent a subset of the change data.	**change-only data, differential update**
Digital image model (Digitales Bildmodell)	A digital image model is a model for storing image data, e.g. digital orthophotos.	**digital image model**
Digital Terrain 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-referenced elementary objects (REO) Spatially-referenced elementary objects are to be formed if, in addition to technical properties, geometric or topological properties are to be demonstrated. Non-spatially-referenced elementary objects (NREO) Non-spatially-referenced elementary objects are to be formed if, in addition to technical properties, no geometric or topological properties can be demonstrated. → See also "Composed objects (ZUSO)" Three dimensional space referenced elementary object (REO_3D) Three dimensional spatially-referenced elementary objects are to be formed when, in addition to technical properties, geometric or topological properties, inclusive of the third dimension, are to be demonstrated.	**elementary objects**
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 is application-specific data of a technical user, e.g. managing data or customer data of a utility company. This 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 (basis data) and the technical data in the form of references. This linking can take place in the spatially-referenced 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**

Fachobjekt	A feature is an 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. → Objekt 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		**Geodetic control station**
Fortführungs-auftrag	The revision order is a feature type in which one or more revision cases are brought together in an entity. It controls the data updating process for all database objects.	**Revision 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 is official geodata that describes the landscape (topology), sites and the buildings in the unified geodetic spatial reference, independent of application. Geobasis data is examined and prepared by the state surveying authority. It fulfils the function of the basis data for geotechnical data.	**(geographic) reference data**
Geodaten	Geographic data (geodata) is data that refers to spatial objects in relation to the earth's body.	**Geographic data**
Geodaten-bestand	Geographic database comprises the totality of geographic data that can be stored in a database.	**Geographic database**
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 inventory refers 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 inventory**
Historisierung	Historicization describes the creation of the last version (expiry) of a feature.	**Histricization**

Identifikator	An 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 set or the number of elements of a finite set. In modelling, this is expressed by the range of potential cardinalities. Common range data in the feature catalogue are: 1..1 Occurs precisely once. 1..* Occurs once or more often. 0..1 Occurs never or once. 0..* Occurs never or more often.	**cardinality**
Kartengeometrieobjekt	Map geometry objects are features 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 an (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 is data on data. It describes 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 or her own purposes.	**metadata catalogue**
Metaobjekt-klasse	Metaclasses are defined as a basis on which features are instantiated. A spatially-referenced metaclass (GF_FeatureType from ISO 19109) is used for modelling the basic classes.	**metaclass**
Methode	A method is a function bound to an object. It affects only this object 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	A conceptual schema language provides illustrative and/or textual elements for describing a model. For modelling in the AFIS-ALKIS-ATKIS technical area, the Unified Modelling 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 TC 211 "Geographic Information/Geomatics" (see http://www.isotc211.org/)for geoinformation. These documents pass through various stages of maturity. The final stage is publishing the "International Standard". For more information: See http://www.iso.ch/.	**de-jure standards**
Nutzer-spezifische Bestandsdaten-aktualisierung	Operation used to update secondary databases with change-only or change data	**user-specific updating 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. → Fachobjekt a discrete entity with a well-defined boundary and identity that encapsulates state and behaviour; an instance of a class	**object**
Objektart	Objects are classified according to various feature 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 type 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 type catalogue**
Objektbehälter	The feature version container forms a technical bracket around the various versions of an object through which it passes during the course of its life. By "bracketing" the versions within a container for feature versions, the technical view of the object remains in place.	**feature versions container**

Objekt-identifikator	→ Identifikator	**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**
Output catalogue	The output catalogue specifies the method of preparing and outputting the data and extracts from AFIS, ALKIS and ATKIS to the user.	**output catalogue**
Portrayal catalogue (Signaturen-katalog)	A portrayal catalogue contains rules 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 adjusted to the respective target scale.	**portrayal catalogue**
Präsentations-objekt	Presentation objects are spatially-referenced elementary objects that supplement the features with details on representing text and portrayals. All texts and portrayals are defined that cannot be fully-automatically generated and positioned for a particular target scale. Presentation objects should be defined in the feature catalogue on which the area is based (e.g. ATKIS-basis-OK).	**presentation object**
Primärnachweis	The primary database is the original database managed by the thematic relevant location (data manager).	**primary database**
Primary database data	Primary database data is geoinformation of Official Surveying and Mapping in AFIS, ALKIS and ATKIS. It contains the complete description of features, including the data on their cartographic or textual depiction in one or several target scales.	**(geographic) data in 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 Pseudocode 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**
PunktLinien Thema	A point and line theme 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 lines and points positioned exactly in the same locations reciprocally separate 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**
Qualifizierung-sprozess	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 satisfies 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 cut-out of 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 administered 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 *Implementation/Refining*.	**relation**
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.	**schema**
Sekundär-nachweis	The secondary database contains a copy of the entire primary database or parts of same, which are continuously updated. The secondary database is revised through the user-related update of the primary database (NBA).	**secondary database**

Standard	A de-facto standard is a widely accepted and applied instrument. It is usually generated by only *one* institution, i.e. no independent international body is responsible for it. 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). These are 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	→ Basisschema	**subschema**
Thematic/ technical information system (Fachinforma-tionssystem)	System containing as its basis information of a technical nature and geobasis information of the surveying and cadastre authority.	**Technical information system**
Transferprozess	*See GeoInfoDok*	**transfer process**
Updating	Revision (update) means the updating of primary database data. The updating data (data and metadata) involved is transferred to the database by applying suitable methods.	**Update, revision**
URI	Uniform Resource Identifier Character string that points unambiguously to a resource (name, file etc). The location of the resource is not restricted (www, LAN, etc). 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). → Versionierungsschema	**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. → Objektbehälter → Versionierung	**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 Mark-up Language). On the basis of the structures stipulated in the XML Schema, XML documents can be created for the exchange of data. See 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.	**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**

10.2 List of Abbreviations Used

Abbreviation	**Long version (original German long text is provided for clarification of abbreviations)**
AdV	Working Committee of the Surveying Authorities of the States of the Federal Republic of Germany
AFIS	Official Fixed Point Information System (Amtliches Festpunktinformationssystem)
ALB	Automated Real Estate Register (Automatisiertes Liegenschaftsbuch)
ALK	Automated Real Estate Map (Automatisierte Liegenschaftskarte)
ALKIS	Official Real Estate Cadastre Information System (Amtliches Liegenschaftskataster Informationssystem)
ATKIS	Official Topographic Cartographic Information system (Amtliches Topographisch-Kartographisches Informationssystem)
ATS	Abstract Test Suite
BKG	Federal Agency for Cartograpy and Geodesy (Bundesamt für Kartographie und Geodäsie)
CD	Committee Draft
CityGML	City Geography Mark-up Language
CRS	Coordinate Reference System
CSL	Conceptual Schema Language
DB	Database

DBM	Digital image model (Digitales Bildmodell)
DGM	Digital Terrain Model (Digitales Geländemodell)
DLKM	Real Estate Cadastre Model
DLKM3D	Three dimensional Real Estate Cadastre Model
DLM	Digital Landscape Model (Digitales Landschaftsmodell)
DOP	Digital orthophoto
DTD	Document Type Definition
DTK	Digital topographic map
DXF	Data exchange format
FIS	Thematic/technical information system (Fachinformationssystem)
GeoInfoDok	Documentation on the Modelling of Geoinformation of Official Surveying and Mapping
GIS	Geoinformationssystem
GML	Geography Mark-up Language
ID	Identifier
IFC	Industry Foundation Classes (Standard for digital building description model)
INSPIRE	Infrastructure for Spatial Information in Europe
ISO	International Organization for Standardization
LoD	Level of detail
NAS	De-jure based data exchange interface (Normbasierte Austauschschnittstelle)
NBA	User-related update of primary database (Nutzerbezogene Bestandsdatenaktualisierung)
NREO	Non-spatially-referenced elementary object (Nicht raumbezogenes Elementarobjekt)
OGC	Open Geospatial Consortium
OK	Objektartenkatalog
REO	Spatially-referenced elementary object
SK	Portrayal catalogue (Signaturenkatalog)
TC	Technical Committee
TK	Topographic map
UML	Unified Modelling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
UUID	Universally Unique Identifier
XML	Extensible Mark-up Language
ZUSO	Composed object (Zusammengesetztes Objekt)

10.3 List of Figures