

ForestGML zur Waldbeschreibung

Eine Modellierungssprache für forstspezifische Sach- und Geodaten

Ein KWH4.0-Standpunkt

30.11.2020

Kompetenzzentrum Wald und Holz 4.0
c/o RIF Institut für Forschung und Transfer e.V. (Projektkoordination)
Joseph-von-Fraunhofer-Straße 20
D-44227 Dortmund
www.kwh40.de

Kontakt

Kompetenzzentrum Wald und Holz 4.0
 c/o RIF Institut für Forschung und Transfer e.V. (Projektkoordination)
 Joseph-von-Fraunhofer-Straße 20
 D-44227 Dortmund
 www.kwh40.de

Ansprechpartner: Dipl.-Ing. Frank Heinze
 Tel. +49 (0) 231 9700-781
 frank.heinze@rt.rif-ev.de

Verantwortlicher Autor: Dr. Martin Hoppen, MMI

Autoren



RIF Institut für Forschung und Transfer e.V. (Kordinator)
 Geschäftsführer: Dipl.-Inf. Michael Saal
 Joseph-von-Fraunhofer Str. 20, 44227 Dortmund



Institut für Mensch-Maschine-Interaktion (MMI), RWTH Aachen
 Institutsleiter: Prof. Dr.-Ing. Jürgen Roßmann
 Ahornstraße 55, 52074 Aachen

Förderhinweis

Dieses Vorhaben wird gefördert durch das Land Nordrhein-Westfalen unter Einsatz von Mitteln aus dem Europäischen Fonds für regionale Entwicklung (EFRE).



EFRE.NRW
 Investitionen in Wachstum
 und Beschäftigung



EUROPÄISCHE UNION
 Investition in unsere Zukunft
 Europäischer Fonds
 für regionale Entwicklung

Version	Datum	Seiten	Änderungen
0.9	12.11.2020	Alle	Interner Entwurf
1.0	30.11.2020	Alle	Erste veröffentlichte Version

ForestGML zur Waldbeschreibung

Die Grundidee der **Forest Geography Modeling Language (ForestGML)** ist die Bereitstellung einer einheitlichen Modellierungssprache bzw. -sprachfamilie zum standardisierten Austausch und zur konsistenten Verwaltung forstlicher Sach- und Geodaten in einem übergreifenden objektorientierten Datenmodell. Die Entwicklung der Sprache begann bereits im Rahmen der Projektreihe „Virtueller Wald“. Die Motivation war das Fehlen eines Sprachstandards für Inventurdaten (Bestandes-, Stichproben- und Einzelbauminventur), einheitlich für Sach- und Geodaten. Ziel war eine GML-basierte Sprache, woraus der Name ForestGML in Anlehnung an CityGML¹ stammt. (Noch) handelt es sich bei ForestGML nicht um einen offiziellen Standard, allerdings wird es bereits in drei Landesforstverwaltungen (RLP, NRW, SL) im Bereich der Forsteinrichtung eingesetzt. Das darunterliegende GML²-Format ist internationaler Standard, so dass z.B. die Grundlagen für standardisierten Datenaustausch z.B. per WFS³ gegeben sind.

Im Kontext von **Wald und Holz 4.0** bildet ForestGML die Grundlage zur Detail-Beschreibung von Waldbeständen. Sie ist nicht zu verwechseln mit der **Forest Modeling Language 4.0 (ForestML 4.0)**⁴ zur Modellierung von WH4.0-Dingen, also WH4.0-Komponenten (Assets wie Forstmaschinen, Sägewerke aber auch der Wald und Personen, zusammen mit dem jeweiligem Digitalen Zwilling (DZ)), WH4.0-Diensten (Softwaredienste) und WH4.0-Mensch-Maschine-Schnittstellen (Apps, Desktop-, Web-Anwendungen). In diesem Zusammenhang bildet ForestML 4.0 das grundlegende Format zur Beschreibung des DZ Wald während im ForestGML-Format die detaillierten Inventurdaten abgelegt werden (siehe Kapitel 5).

Im vorliegenden KWH4.0-Standpunkt soll nun die Architektur und Nutzung von ForestGML zur Waldbeschreibung vermittelt werden. Im Anhang dieses Standpunkts wird auf die technische Spezifikation des Basisschemas (**ForestGML Base**) sowie der Umsetzungen für die Bestandesinventur (BI) für die Verhältnisse in den Bundesländern Nordrhein-Westfalen (**ForestGML BI NRW**) und Rheinland-Pfalz (**ForestGML BI RLP**) verwiesen. Der Fokus liegt zunächst auf der Bestandesinventur. Die Ausprägungen in die Richtungen Stichprobeninventur und Einzelbauminventur werden in folgenden Versionen dieses Standpunkts beschrieben.

1 Architektur von ForestGML

Abbildung 1-1 zeigt den Aufbau von ForestGML für die Waldbeschreibung. Das Basisschema **ForestGML Base** stellt die grundlegenden Strukturen bereit, die noch unabhängig von konkreten Anwendungen oder Inventurmethode sind. Beispiele sind Strukturen zur Modellierung hierarchisch strukturierter Objekte mit Flächen-, Linien-, oder Punktgeometrie. Auf dem Basisschema basieren die verschiedenen Basis-Fachschemata für die Stichprobeninventur **ForestGML SP** (Sample Plot), Einzelbauminventur **ForestGML ST** (Single Tree) und Bestandesinventur **ForestGML SI** (Stand Inventory). Sie erweitern das Basisschema dazu um Modellierungskonzepte u.a. für Bäume, Stichprobenpunkte oder Baumartenzeilen. Diese sind nur eingeschränkt für die direkte Nutzung vorgesehen, sondern dienen in erster Linie als einheitliche Grundlage für die konkreten Fachschemata. Für die Stichprobeninventur wurden hier Prototypen konkreter Fachschemata für die Landeswaldinventur in NRW (**ForestGML SP LWI-NRW**) oder für die Bundeswaldinventur (**ForestGML SP BWI**) entwickelt. Für die Bestandesinventur stehen Fachschemata für NRW (**ForestGML BI NRW**) und Rheinland-Pfalz (**ForestGML BI RLP**) bereit,

¹ City Geography Markup Language <https://www.opengeospatial.org/standards/citygml>

² Geography Markup Language <https://www.opengeospatial.org/standards/gml>

³ Web Feature Service <https://www.ogc.org/standards/wfs>

⁴ Siehe KWH4.0-Standpunkt „Forest Modeling Language 4.0“ auf <https://www.kwh40.de/veroeffentlichungen/>

die dort bereits im praktischen Einsatz der Landesforsten sind. Das Basisschema basiert wiederum auf dem GML-Basisschema der OGC. Dadurch wird insbesondere ein standardkonformer Austausch über WFS[-T]⁵ und eine standardkonforme Datenverarbeitung (z.B. zur Kartendarstellung über SLD⁶) sichergestellt.

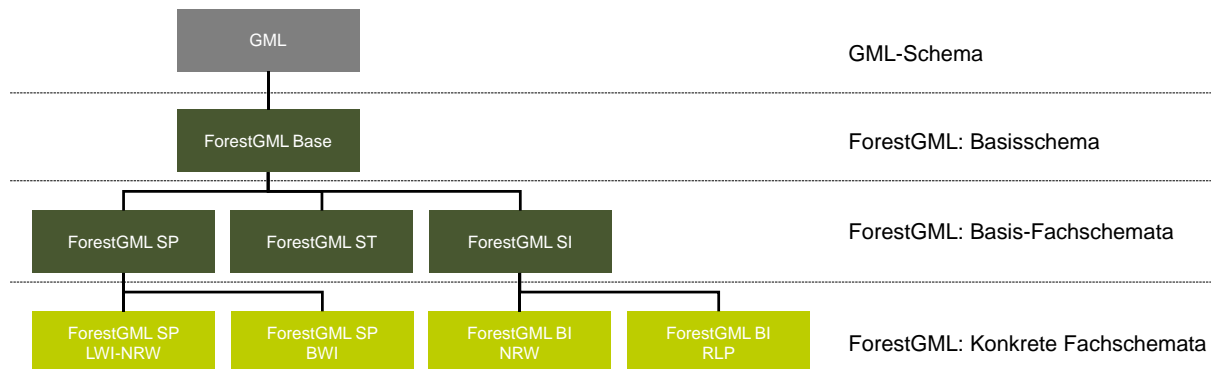


Abbildung 1-1: Aufbau der ForestGML-Sprachfamilie

2 Grundlagen der Datenmodellierung

Modellierungssprachen wie ForestGML werden über ein **Datenmodell** spezifiziert. Dabei gibt es verschiedene Ebenen, auf denen Konzepte und Ideen beschrieben werden können (Abbildung 2-1).

Ein **konzeptuelles Datenmodell** beschreibt die Konzepte und Ideen auf abstrakter Ebene. Dabei liegt der Fokus auf der Problem-Domäne und nicht auf der Umsetzung, d.h. ohne Bezug zu einer konkreten Lösungstechnologie. Unabhängig von einer konkreten technischen Realisierung gibt es das konzeptionelle Datenmodell nur genau einmal. Für eine Formalisierung der Repräsentation werden Modellierungssprachen wie die Unified Modeling Language⁷ (UML) oder Entity-Relationship-Diagramme (ERD) verwendet.

Aus dem konzeptionellen können verschiedene **logische Datenmodelle** für verschiedene Umsetzungsvarianten abgeleitet werden. Dabei werden weiterhin dieselben Konzepte beschrieben, nur für unterschiedliche Lösungsansätze bzw. Zielsysteme. Beispiele sind die verschiedenen Ausprägungen von ForestGML in OPC UA⁸, in XML Schema Description (XSD)⁹ oder für relationale Datenbanken. Auf dieser Ebene können weiterhin allgemeine Modellierungssprachen wie UML oder ERD (mit um Implementierungsdetails angereicherten Modellen) oder domänenspezifische Sprachen wie SQL verwendet werden.

Aus logischen können wiederum **physikalische Datenmodelle** abgeleitet werden. Hierbei findet der Übergang in konkrete Softwaresysteme, Dateiformate oder Datenbanksysteme statt. Beispiele sind ForestGML im XML-Format für OPC UA-Informationsmodelle („Nodesets“), eine XSD-Datei für ForestGML oder eine konkrete Oracle-Instanz mit Tabellen für ForestGML.

Insgesamt folgt daraus, dass ein einheitliches Datenmodell auf konzeptueller Ebene in den verschiedensten Zielsystemen unabhängig von der konkret gewählten technischen Umsetzung unterschiedlich und damit bedarfsgerecht umgesetzt werden kann.

⁵ Web Feature Service [Transactional] <https://www.opengeospatial.org/standards/wfs>

⁶ Styled Layer Descriptor <https://www.opengeospatial.org/standards/sld>

⁷ <https://www.omg.org/spec/UML/>

⁸ <https://opcfoundation.org/>

⁹ <https://www.w3.org/standards/xml/schema>

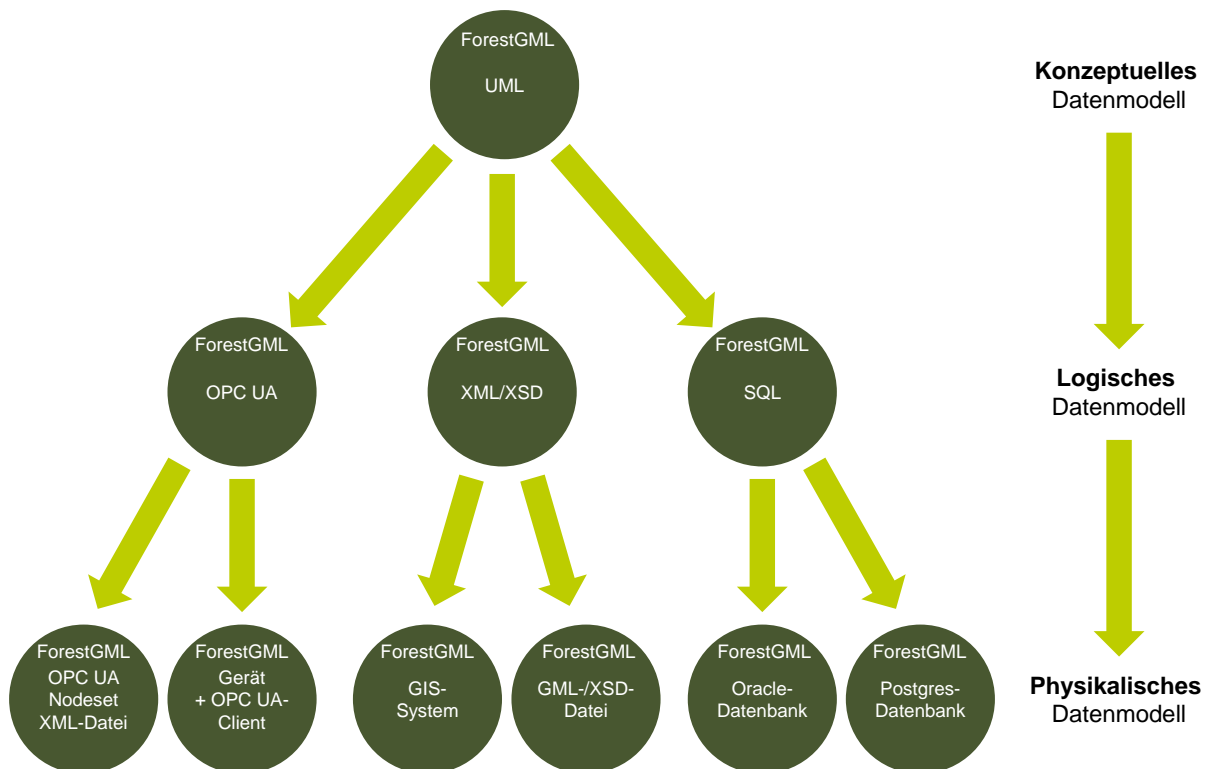


Abbildung 2-1: Veranschaulichung des Unterschieds zwischen konzeptuellen, logischen und physikalischen Datenmodellen am Beispiel ForestGML

3 Zentrale Modellierkonzepte in ForestGML

Im Folgenden werden zentrale Modellierkonzepte des konzeptuellen Datenmodells von ForestGML erläutert. Zur graphischen Darstellung wird dabei die UML genutzt. Hierbei werden UML-Diagramme mit diesen Konventionen verwendet:

- Dunkel-Grau: **GML** – Namespace „gml“
- Hell-Orange: **ForestGML Base / SP / ST / SI** – „fgml“
- Grün: **ForestGML BI RLP / NRW** (zentrale Klassen) – „fgmlbinrw“ / „fgmlbirlp“
- Orange: **ForestGML BI RLP / NRW** (Definitionen & Dictionaries) – „fgmlbinrw“ / „fgmlbirlp“

Weitere Sprachkonventionen:

- Basisschema und Basis-Fachschemata sind in Englisch gehalten, um die Verbreitung der Sprache zu unterstützen
- Aktuell umgesetzte konkrete Fachschemata sind auf Deutsch gehalten, um die Anwendungsdomäne möglichst detailgetreu abzubilden
- Die meisten Basisklassen sind abstrakt, d.h. sie können nur indirekt durch Vererbung in einem konkreten Fachschema für Daten genutzt werden
- Abstrakte Basisklassen tragen den Präfix „Abstract“ im Namen

Im Folgenden werden zentrale Aspekte des Basisschemas und der Basis-Fachschemata erklärt und anhand konkreter Beispiele aus den konkreten Fachschemata für die Bestandesinventur in NRW und RLP erläutert.

3.1 Zentrale Basisklassen

Als GML-konformes Datenmodell basieren die vier zentralen Basisklassen von ForestGML direkt auf GML-Basisklassen (Abbildung 3-1).

- **fgml::AbstractForestGMLFeature** Objekte mit Raumbezug, z.B. Waldorte in RLP
- **fgml::AbstractForestGML** Objekte ohne Raumbezug, z.B. Baumartenzeilen
- **fgml::Definition** Legendeinträge als Schlüssel-Wert-Paare, z.B. Baumart 80→Fichte in NRW
- **fgml::Dictionary** für Legenden als Sammlungen von Legendeinträgen, z.B. Legende Baumart

Sämtliche ForestGML-Klassen basieren direkt oder indirekt auf einer dieser vier Klassen.

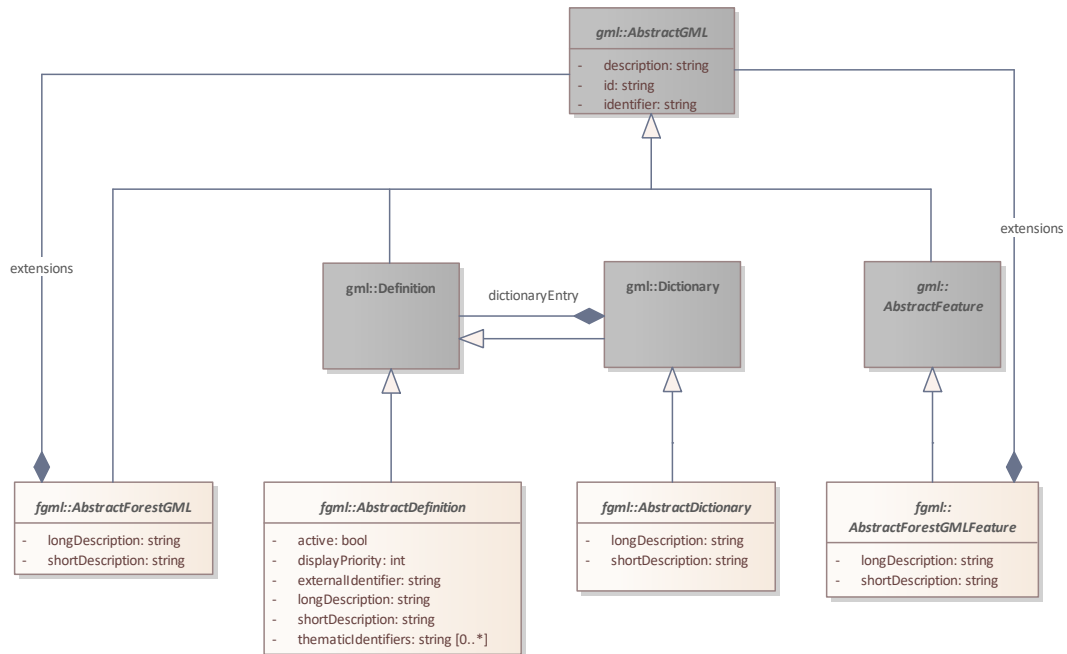


Abbildung 3-1: Die vier zentralen Basisklassen von ForestGML

3.2 Grundaufbau von Datenmodellen

Abbildung 3-2 zeigt den Grundaufbau von ForestGML-konformen Datenmodellen wie ForestGML BI NRW oder ForestGML BI RLP. Die erste ForestGML-Modellierungsregel lautet, dass ein zugehöriger ForestGML-konformer Datensatz stets ein eindeutiges Wurzelobjekt (bzw. auf Ebene des Datenmodells die zugehörige Klasse) haben muss, mit der direkt oder indirekt jedes Datenobjekt verbunden ist (siehe „ForestGML-Modellierungsregel 1“). Dazu dient die Basisklasse **fgml::AbstractDataModelRoot** bzw. deren Konkretisierung für Inventurdaten **fgml::AbstractInventory** und die weiteren Spezialisierungen für Bestandesinventurdaten (**fgml::AbstractStandInventory**) oder Stichprobeninventuren (**fgml::AbstractSamplePlotInventory**). Davon leiten sich dann wiederum die konkreten Spezialisierungen z.B. für die Bestandesinventur in RLP (**fgmlbirlp::Bestandesinventur**) ab.

ForestGML-Datensätze haben immer ein eindeutiges Wurzelobjekt, dessen Klasse von fgml::AbstractDataModelRoot erbt.

ForestGML-Modellierungsregel 1

Ein solcher Wurzelknoten für Datenmodelle verfügt dann insbesondere über eine Sammlung von Legenden (**fgml::DictionaryLibrary**) und über eine Liste sogenannter Datenordner (**fgml::AbstractFolder** und Spezialisierungen). In diesen Datenordnern befinden sich die maßgeblichen Daten des jeweiligen Datenmodells, z.B. Stichprobenpunkte im Fall einer Stichprobeninventur (**fgml::AbstractFolder-SamplePlotInventory** mit **fgml::AbstractSamplePlot**).

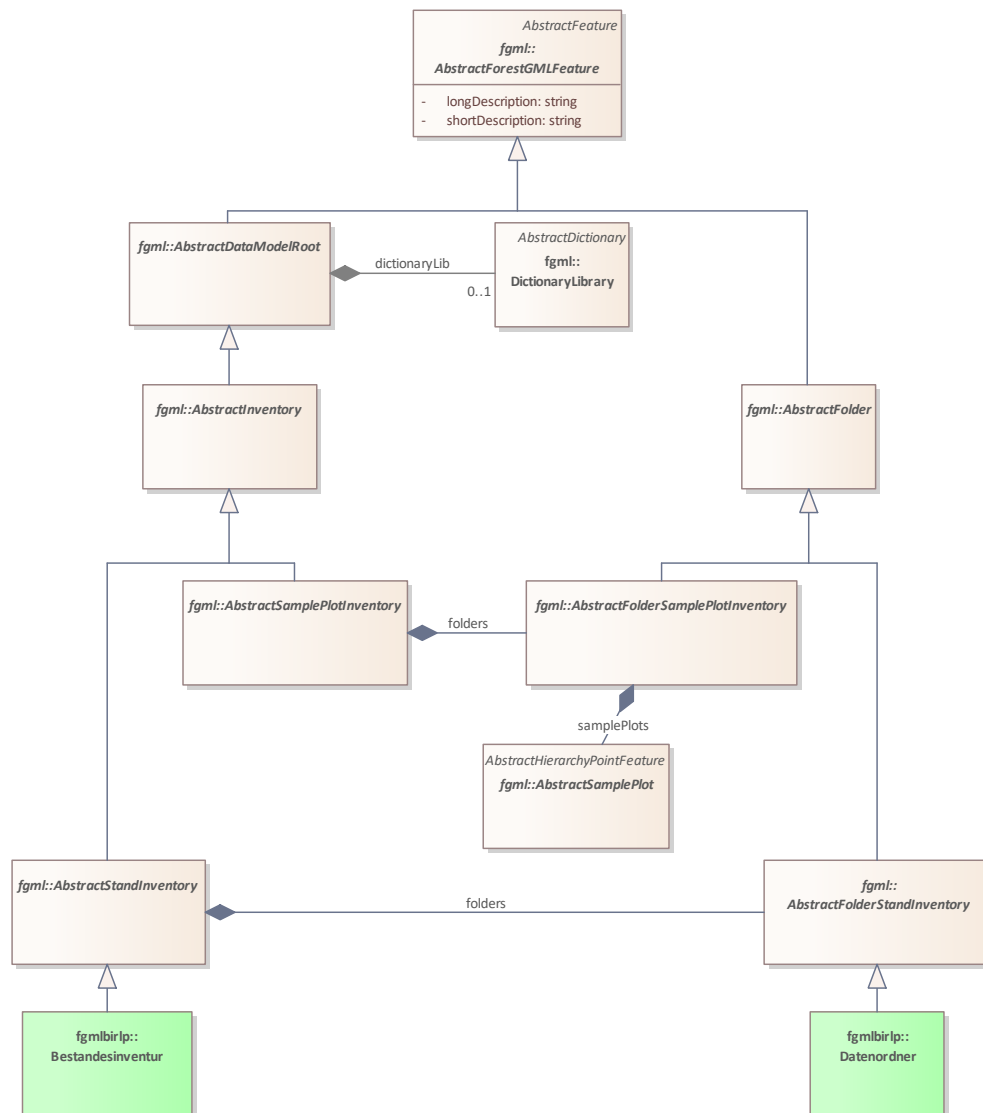


Abbildung 3-2: Grundaufbau von ForestGML-konformen Datenmodellen

3.3 Legenden

Zentraler Bestandteil jedes ForestGML-Datenmodells ist eine Sammlung von Legenden (oft auch „Lookups“ oder „Codelisten“ genannt). In ForestGML basieren diese auf dem von GML übernommenen „Dictionary-Definition“-Prinzip und erben von den entsprechenden Basisklassen. Die Klasse **fgml::DictionaryLibrary** dient dabei der Zusammenstellung sämtlicher Legenden eines Datenmodells und damit sämtlicher Konkretisierungen von **fgml::AbstractDictionary** für die verschiedenen Anwendungsfälle; in der Abbildung beispielhaft für eine Legende „Geologie“ (**fgmlbirlp::GeologieDictionary**) dargestellt. Konvention ist hier die Nutzung des Suffixes „Dictionary“ (ForestGML-Modellierungsregel 2). Die entsprechenden Einträge nutzen dann im Beispiel die Klasse **fgmlbirlp::Geologie**, die von **fgml::AbstractDefinition** erbt. Zentrale Legenden wie für die Baumarten werden dabei bereits über spezialisierte Basisklassen auf Ebene des Basisschemas definiert und dann auf Ebene jedes konkreten Fachschemas final definiert (**fgml::TreeSpecies** und **fgmlbirlp::Baumart**). Dies wird immer dann genutzt, wenn die entsprechenden Konzepte bereits auf Eben der (Fach-)Basisschemata genutzt werden. Im allgemeinen wird für jede Legende eine konkrete Spezialisierung von **fgml::AbstractDefinition** genutzt.

3.4 Zentrale Feature-Klassen und Hierarchie

Zentrale Klassen zur Beschreibung von Objekten mit (unmittelbarem) Raumbezug, sogenannter Features, sind in Abbildung 3-4 dargestellt. ForestGML-basierte Datenmodelle sehen eine hierarchische Strukturierung der Daten vor, wie sie in Inventurdaten als sogenannte „Forsthierarchie“ zu finden ist (z.B. Betrieb-Abteilung-Unterabteilung-Bestandeseinheit in NRW oder Betrieb-Distrikt-Abteilung-Waldort-Befundeinheit in RLP). ForestGML bildet diese Variabilität verschiedener Hierarchisierungen über die Basisklasse ***fgml::AbstractHierarchyFeature*** bzw. ihre Spezialisierungen für Linien (***fgml::AbstractHierarchyCurveFeature***), Punkte (***fgml::AbstractHierarchyPointFeature***) und Flächen (***fgml::AbstractHierarchySurfaceFeature***) ab.

Aufbauend auf diesen technischen Basisklassen wurden fachliche Basisklassen für Einzelbäume (***fgml::AbstractTree***), Stichproben (***fgml::AbstractSample***) oder Probenkreise (***fgml::AbstractSamplePlot***) definiert, die von ***fgml::AbstractHierarchyPointFeature*** (direkt/indirekt) erben. Zur Beschreibung von Flächenobjekten mit Standortsbeschreibung (z.B. Unterabteilung in NRW oder Waldort in RLP) wird die Basisklasse ***fgml::AbstractSite***, zur Beschreibung von Bestandesobjekten (z.B. Bestandeseinheit in NRW oder Befundeinheit in RLP) die Klasse ***fgml::AbstractStand*** genutzt. Beide erben von ***fgml::AbstractHierarchySurfaceFeature***. Die Basis-Eigenschaft ***relativeID*** beschreibt dabei die Kennung relativ zum hierarchisch übergeordneten Objekt (z.B. die Nummer einer Stichprobe bezogen auf ihren Probekreis oder die Nummer einer Abteilung bezogen auf ihren Betrieb).

Bei der Modellierung von Forsthierarchien mit ForestGML umfassen Bestände Schichten, die wiederum die eigentlichen Bestandesinventurattribute („Baumartenzeilen“) umfassen. Dazu wird zunächst die Basisklasse ***fgml::AbstractStrataContainer*** genutzt, die eine Ansammlung von Schichten (***fgml::AbstractStratum***) als Sachdatenobjekte (vgl. Abschnitt 3.5) beschreibt, die wiederum die zugehörigen Bestandesinventurattribute (***fgml::AbstractStandAttributSet***) enthalten. Eine zentrale Ausprägung solcher „Schichten-Container“ sind dann wiederum Bestände (***fgml::AbstractStand***).

Insgesamt darf nur eine einzige hierarchische Aufteilung eine Existenzabhängigkeit bedeuten (UML composition; ausgefüllter Diamant), parallele Hierarchien dürfen nur ohne Existenzabhängigkeit gebildet werden (UML shared aggregation; nicht-ausgefüllter Diamant). Im Umkehrschluss gilt somit als allgemeine Modellierungsregel in ForestGML-basierten Datenmodellen, dass es von jedem Objekt genau einen Pfad über Kompositionsbeziehungen zum Wurzelobjekt des Datenmodells geben darf (ForestGML-Modellierungsregel 3).

In ForestGML-konformen Datensätzen hat jedes Objekt genau einen Pfad über Kompositionsbeziehungen zum Wurzelobjekt.

ForestGML-Modellierungsregel 3

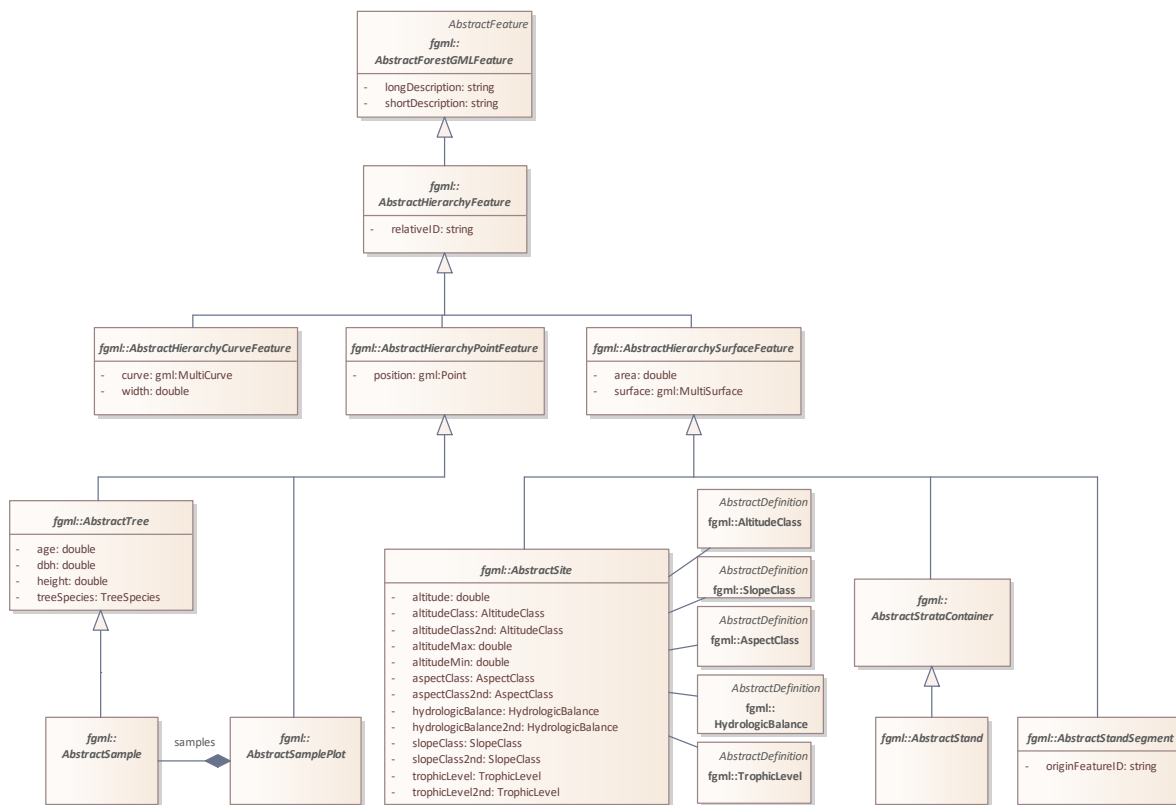


Abbildung 3-4: Zentrale ForestGML-Basisklassen für Objekte mit Raumbezug (Feature)

Abbildung 3-5 zeigt ein konkretes Beispiel für eine ForestGML-basiert modellierte Forsthierarchie aus dem Fachschema RLP. Die *fgmlbirlp::Bestandesinventur* aggregiert *fgmlbirlp::Datenordner*. Diese wiederum bilden zunächst eine Aufteilung der Bestandesinventurdaten in *fgmlbirlp::Betrieb*-*fgmlbirlp::Distrikt*-*fgmlbirlp::Abteilung*-*fgmlbirlp::Waldort* ab. Neben dieser primären, Besitz-orientierten Hierarchie gibt es parallel verlaufende hierarchische Einteilungen über *fgmlbirlp::Funktions-einheit* bzgl. der Bewirtschaftung und *fgmlbirlp::Organisationseinheit* bzgl. hoheitlicher Verhältnisse oder generisch in *fgmlbirlp::Waldortgruppe*.

3.5 Zentrale Sachdaten-Klassen

Neben den zuvor genannten, zentralen Basisklassen zur Beschreibung von Feature-Objekten mit Raumbezug umfasst ForestGML auch Basisklassen für reine Sachdatenobjekte (Abbildung 3-7). Zentral definiert dabei die Klasse **fgml::AbstractAttributeSet** ein Sachdatenobjekt als Zusammenstellung zusammengehöriger Eigenschaften. Eine direkte Spezialisierung davon stellt **fgml::AbstractStandAttributeSet** dar, worüber Inventurattribute von Beständen („Baumartenzeilen“) modelliert werden können.

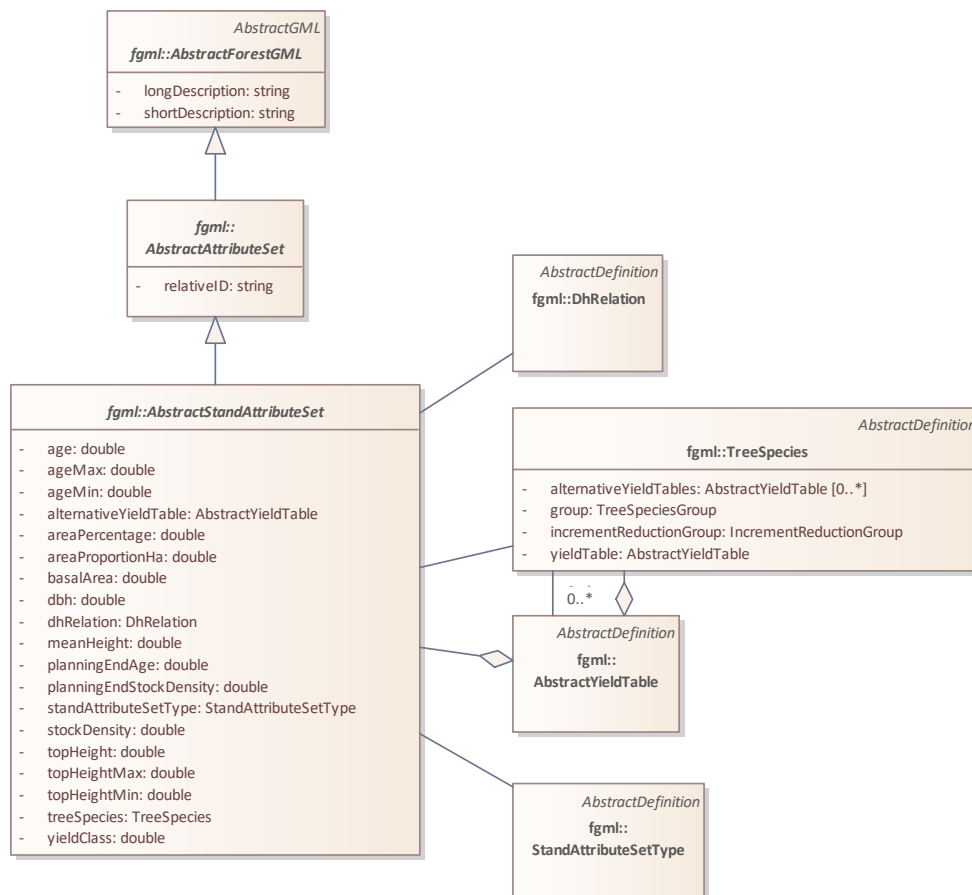


Abbildung 3-7: Zentrale ForestGML-Klassen für Sachdatenobjekte

Abbildung 3-8 zeigt ein Beispiel aus ForestGML BI RLP. Die **fgmlbirlp::Befundeinheit** aggregiert dabei als „Schichten-Container“ Objekte der Klasse **fgmlbirlp::Schicht**. Schichten besitzen über ihre Basisklasse **fgml::AbstractStratum** die Eigenschaft **type**, mit der der konkrete Typ (**fgml::StratumType**) der Schicht als Legende modelliert wird. In einer konkreten Ausprägung dieser Legende können dann Werte wie `0` → Hauptschicht etc. definiert werden. Die **fgmlbirlp::Schicht** setzt sich dann über die Basiseigenschaft **attributes** aus **fgmlbirlp::Baumartenzeilen** zusammen.

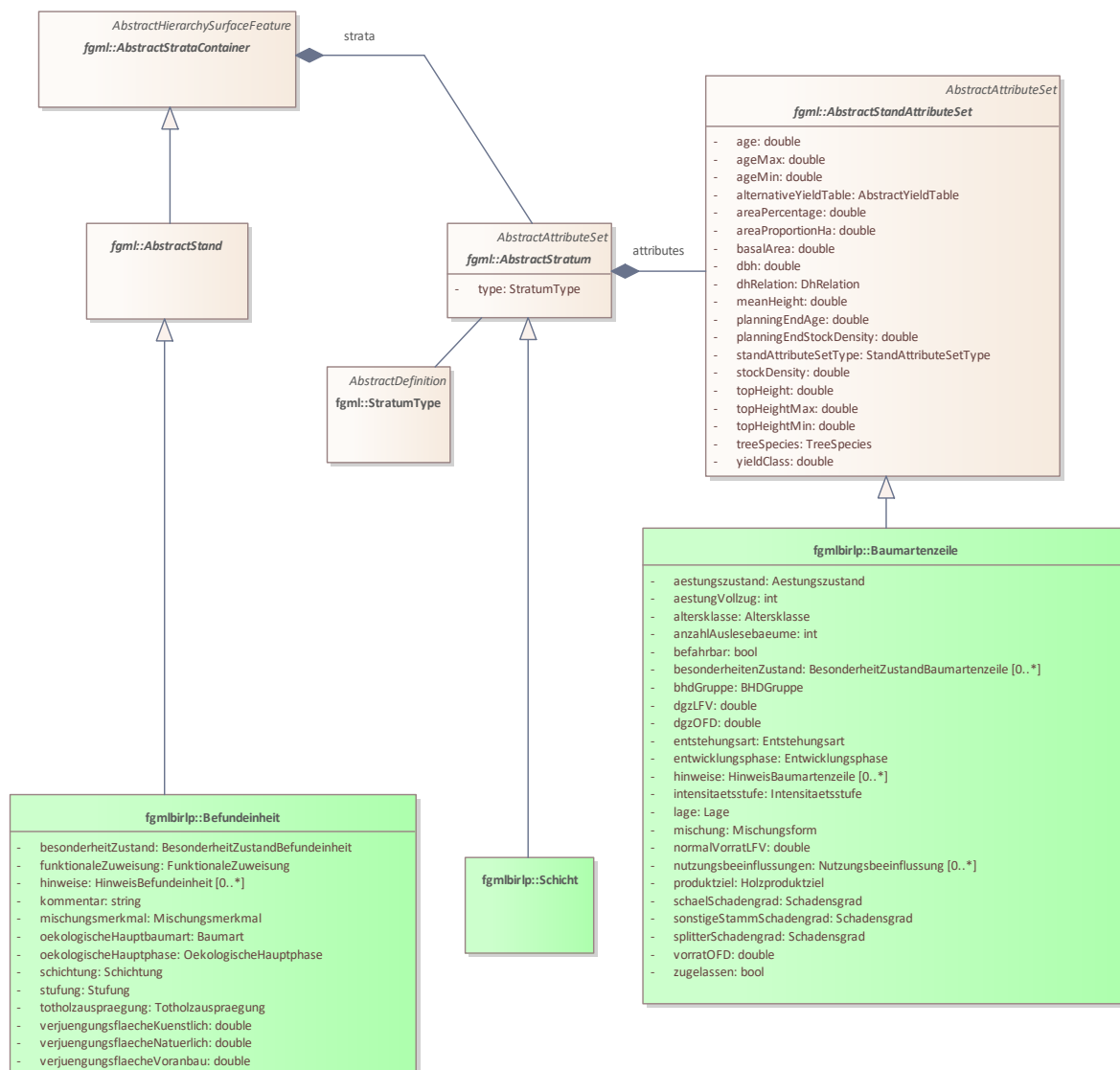


Abbildung 3-8: Beispiel für Sachdatenobjekte: Befundeinheit und Baumartenzeile in RLP

3.6 Prozesssteuerung und Metadaten

Ein weiterer Aspekt von ForestGML ist die Prozesssteuerung. Die Daten sollen nicht nur den Zustand, sondern auch aktive und durchgeführte Prozesse beschreiben. Dazu beschreibt ForestGML sogenannte **Prozesssteuerobjekte** („PSO“; *fgml:AbstractProcessControl*), die in den Fachschemata konkreter für die verschiedenen Prozesse ausgeprägt werden (Abbildung 3-9). Konkrete Beispiele aus RLP sind die Forsteinrichtung (*fgmlbirlp::PSOForsteinrichtung*) oder die Organisationsänderung (*fgmlbirlp::PSO-Organisationsaenderung*). Diese PSOs können jedem Hierarchieobjekt (*fgml:AbstractHierarchyFeature*) zugeordnet werden und beschreiben damit, dass dieses und alle hierarchisch untergeordneten Objekte dem beschriebenen Prozess unterworfen sind oder waren.

PSOs dienen somit sowohl der Beschreibung des Prozesses als auch der Steuerung von Prozessen. Letztere kann darüber gewährleistet werden, dass vor der Verarbeitung von Daten geprüft wird, inwiefern bereits Prozesse aktiv darauf sind. Dazu haben PSOs die Eigenschaft **state** mit der zugehörigen Legende *fgml:ProcessState*, die die Zustände geplant, aktiv und abgeschlossen abbildet.

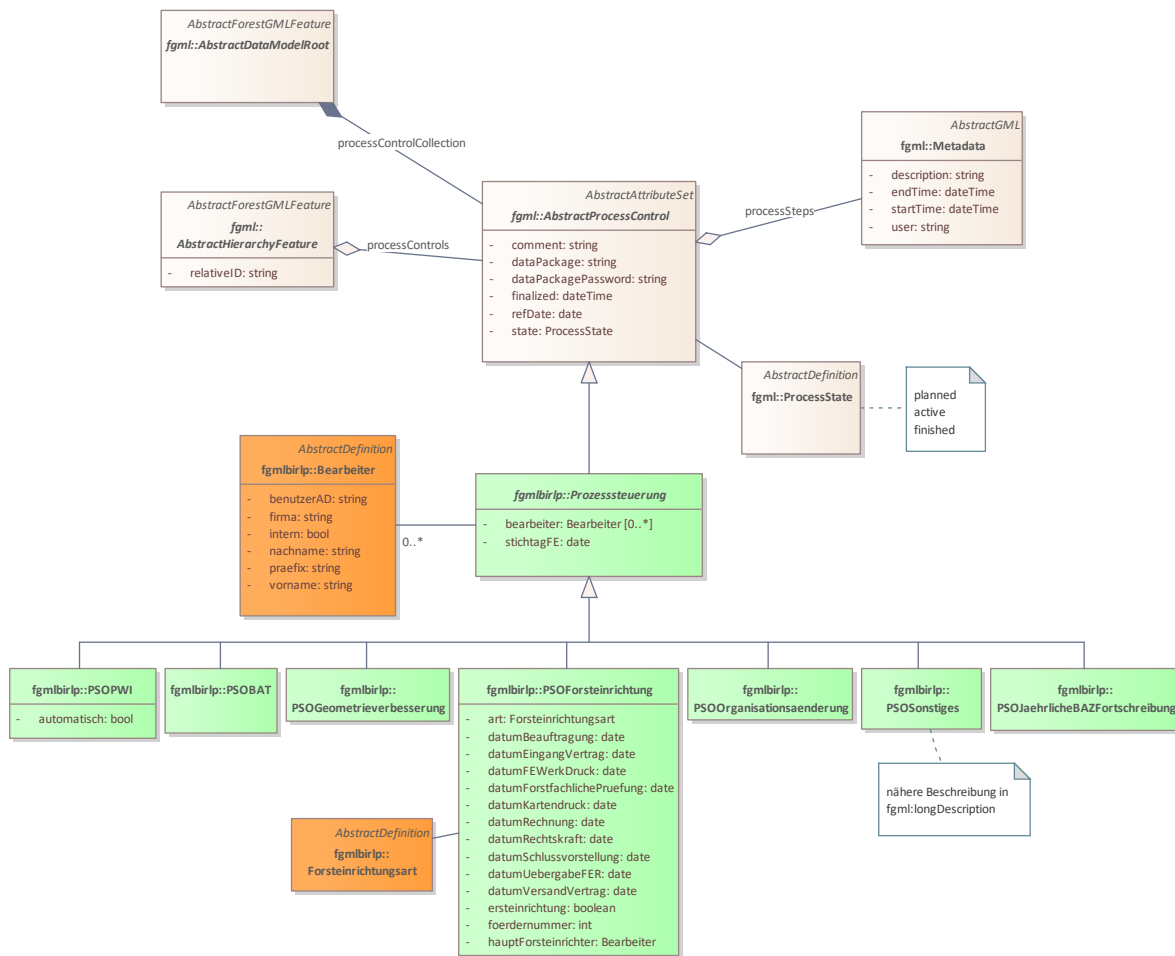


Abbildung 3-9: Prozesssteuerung mit ForestGML

Zusätzlich kann jedes PSO eine Liste von Metadatenätzen (*fgml:Metadata*) umfassen (Abbildung 3-10). Diese dienen als allgemeine Beschreibung von Prozessschritten (*processSteps*) des zugehörigen Prozesses. Dazu können diese mit jedem ForestGML-Objekt verknüpft werden, das im Rahmen eines Prozesses verändert wurde. Die Metadatenobjekte beschreiben Start und Ende eines Teilprozesses, wer ihn durchgeführt hat und was durchgeführt wurde.

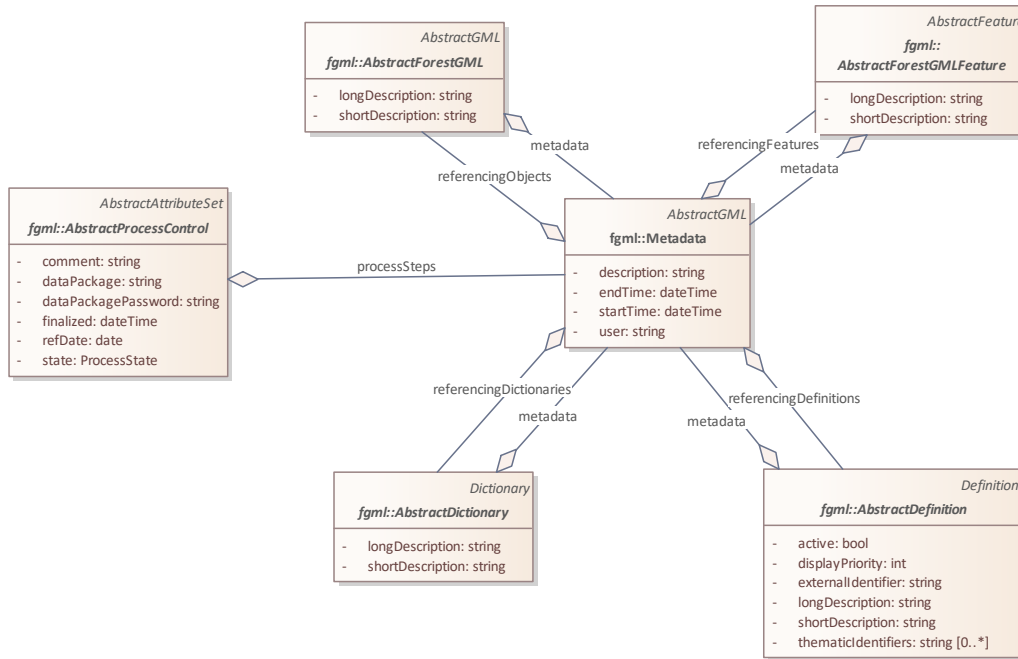


Abbildung 3-10: Metadaten der Prozesssteuerung in ForestGML

3.7 Fällmaßnahmen

ForestGML beschreibt eine Möglichkeit zur Modellierung von Fällmaßnahmen (Abbildung 3-11). Die Basisklasse **fgml::AbstractFelling** erlaubt dazu die Modellierung mit Produktionsdaten (**fgml::ProductionData**) und zugehörigen Einzelbaumobjekten (**fgml::FelledTree**). Diese setzen sich wiederum aus Baumabschnitten (**fgml::Log**) zusammen. Die Produktionsdaten können z.B. über STM-Daten abgebildet werden.

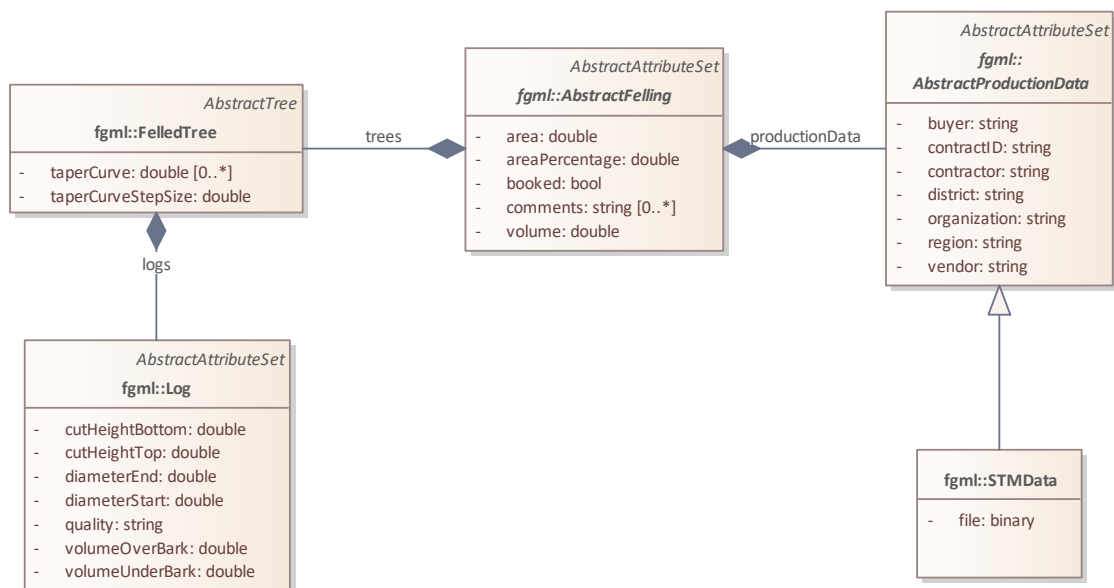


Abbildung 3-11: Modellierung von Fällmaßnahmen mit ForestGML

3.8 GIS-Verschneidung

Neben den Legenden kann jedes ForestGML-konforme Datenmodell zudem über eine Sammlung von GIS-Verschneidungsregeln verfügen (Abbildung 3-12). Dazu umfasst die Wurzelobjektklasse **fgml::AbstractDataModelRoot** eine Liste von Verschneidungsregelsätzen (**fgml::DataExtractionRuleSet**). Diese Regelsätze umfassen alle für eine Featureklasse definierten Regeln (**fgml::DataExtractionRule**), die wiederum generisch aus Schlüssel-Wert-Paaren (**fgml::DataExtractionRuleParameter**) aufgebaut sind. Abhängig von dem die Daten verarbeitenden Programm bzw. Dienst können diese Parameter dann frei gewählt werden, um die Verschneidung der Inventurdaten mit thematischen Vektordaten oder Rasterdaten zur automatischen Ableitung insbesondere von Standortinformationen zu ermöglichen.

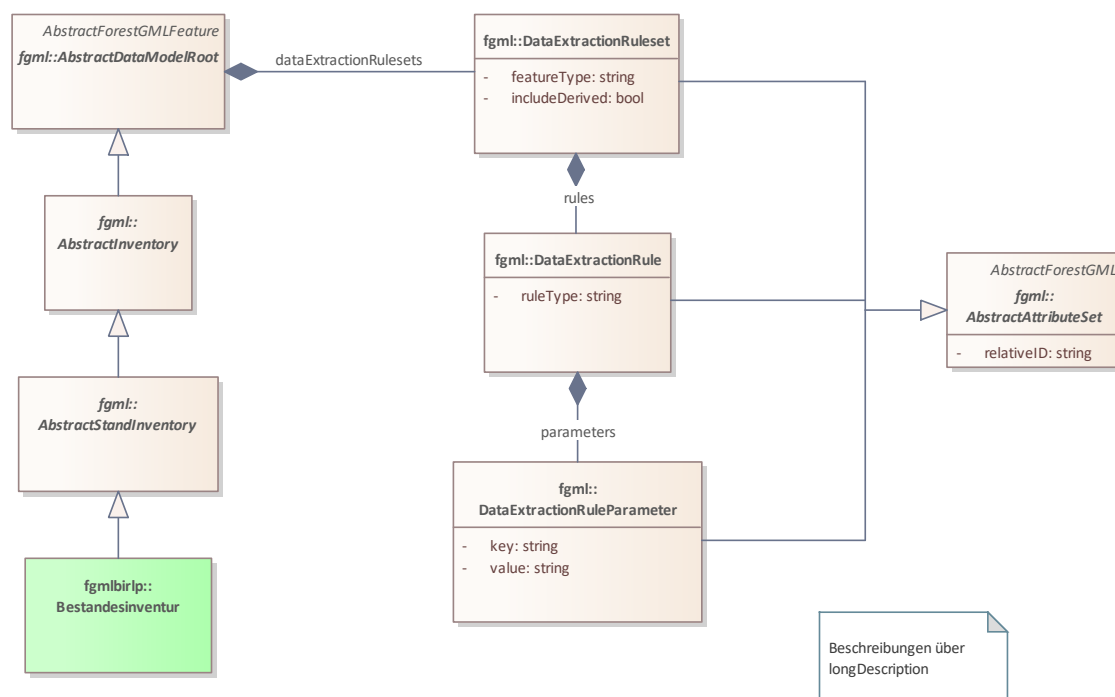


Abbildung 3-12: GIS-Verschneidungsregeln mit ForestGML

Üblicherweise ist das Ergebnis einer Verschneidung ein bestimmter Legendeintrag, z.B. eine bestimmte Wasserhaushaltsstufe, die an eine Unterabteilung (NRW) geschrieben werden soll. Dazu müssen die thematischen Featureobjekte (z.B. aus der BK5) den entsprechenden Legendeinträgen des ForestGML-basierten Datenmodells zugeordnet werden. Zur Identifikation des korrekten Legendeintrags kann die Eigenschaft **thematicIdentifiers** (siehe Abbildung 3-3) von **fgml::AbstractDefinition** genutzt werden, um eine solche externe Identifizierung zu ermöglichen. In das Feld wird dann z.B. bei der entsprechenden Wasserhaushaltsstufe derjenige Code eingetragen, den die entsprechende Stufe in der BK5 aufweist. Im Rahmen einer GIS-Verschneidung kann so der passende Legendeintrag vollautomatisch identifiziert werden.

Soll eine GIS-Verschneidung nicht nur einen einzelnen (häufigsten / dominanten) Wert, sondern sämtliche Paarungen aus auftretenden Werten abbilden, so kann dafür das ForestGML-Konzept der **Vorkommnisse** genutzt werden (Abbildung 3-13). Dazu wird die Basisklasse **fgml::AbstractDefinitionOccurrence** genutzt. In der Basis-Eigenschaft **extent** wird der Umfang beschrieben, mit dem ein gewisser Wert im Rahmen einer GIS-Verschneidung aufgetreten ist (z.B. relativ in Prozent oder absolut in Hektar). Der konkrete Wert wird typischer erst in einer Spezialisierung für die jeweilige Legende (z.B. **fgmlbinrw::HangneigungVorkommnis**) als Eigenschaft **typ** (Nameskonvention) mit dem Datentyp der

jeweiligen Legende (z.B. *fgmlbinrw::Hangneigung*) gespeichert. Im Beispiel für die Hangneigung in NRW können darüber sämtliche Paare aus Häufigkeit einer bestimmten Hangneigungsklasse innerhalb einer Unterabteilung (in Prozent) zusammen mit der zugehörigen Hangneigungsklasse selbst modelliert und über die Eigenschaft *vorkommnisse* des Fachschemas der Unterabteilung zugeordnet werden.

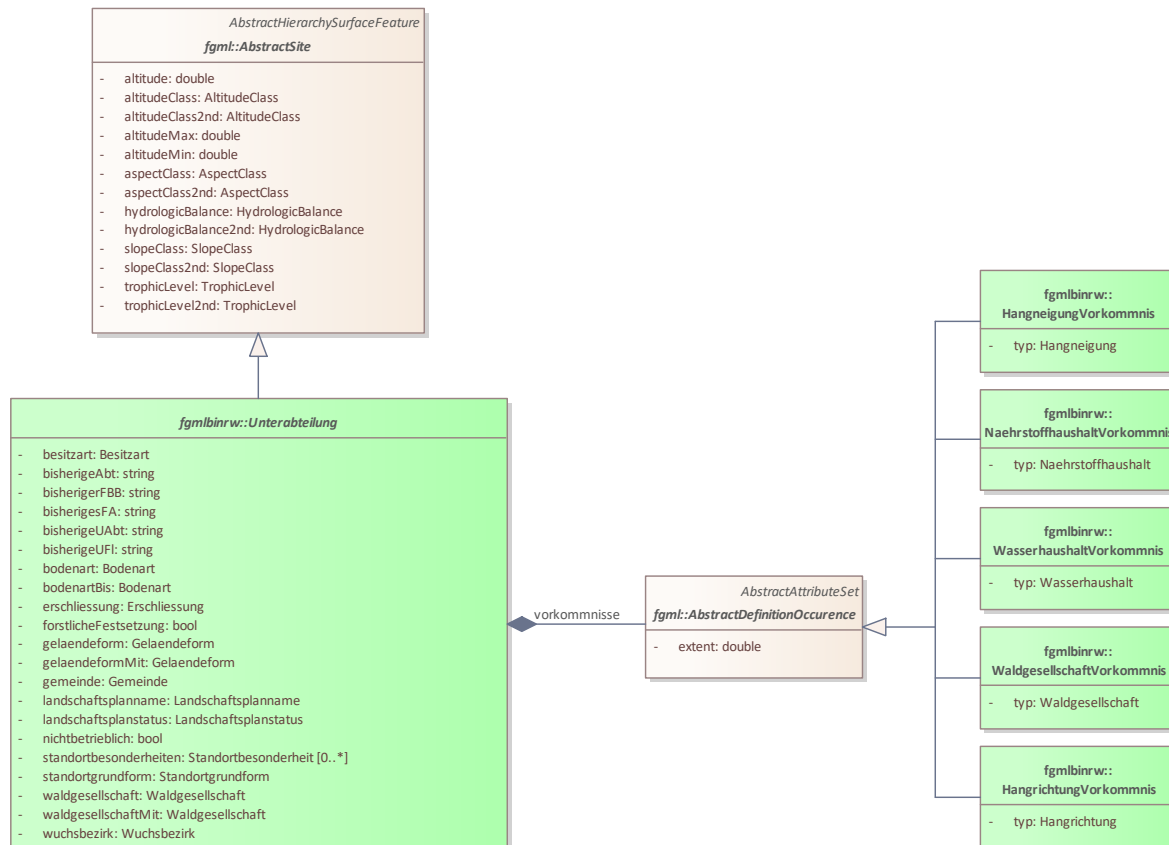


Abbildung 3-13: Beispiele für Vorkommnisse im Datenmodell ForestGML BI NRW

3.9 Plausibilitätstests

Ein weiterer Aspekt sind Plausibilitätstests. ForestGML (Abbildung 3-14) sieht dazu das Konzept eines Testprofils (*fgml::PlausibilityTestProfile*) vor, das wiederum mehrere Tests (*fgml::AbstractPlausibilityTest*) umfasst. Diese können andere Tests als Vorbedingungen (*preconditions*) haben. Zwei Ausprägungen von Tests werden unterschieden. Dies sind zum einen fest in Systeme einprogrammierte Tests (*fgml::PlausibilityTestReference*). Der Testname wird hier über die geerbte Eigenschaft *shortDescription* referenziert. Zum anderen werden OCL¹⁰-basierte Tests (*fgml::PlausibilityTestOCL*) unterstützt, bei denen Testregeln dynamisch in den Stammdaten definiert werden können.

¹⁰ Object Constraint Language <http://www.omg.org/spec/OCL/>

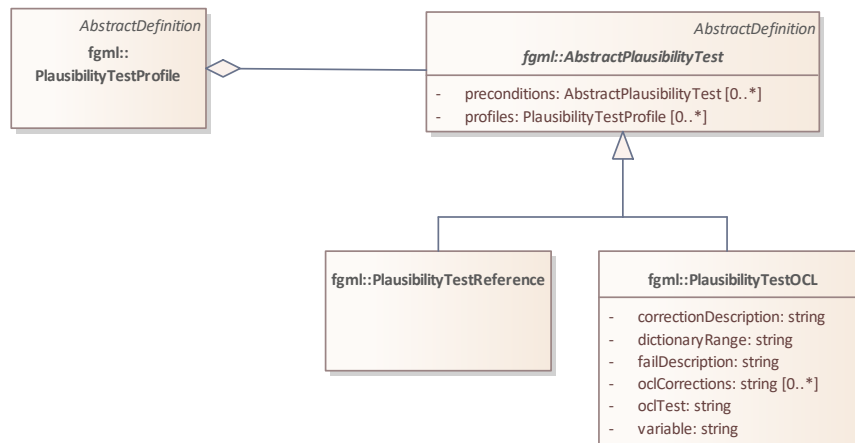


Abbildung 3-14: Modellierung von Plausibilitätstests in ForestGML

3.10 Baumarten und Ertragstafeln

Ein zentraler Aspekt von Inventurdaten im allgemeinen und Bestandesinventurdaten im speziellen sind Baumarten und Ertragstafeln. Daher umfasst ForestGML entsprechende Basisklassen bereits im Fach-Basisdatenmodell. Abbildung 3-15 zeigt diese Strukturen zusammen mit Beispiel-Spezialisierungen zur Nutzung in RLP. Die zentrale Basisklasse *fgml::TreeSpecies* beschreibt Baumarten, die zunächst in verschiedene Baumartengruppen (*fgml::TreeSpeciesGroup*) gruppiert werden können. Für die Kartenbildung kann jeder Baumartengruppe eine Hauptfarbe (*mainColor* vom Typ *fgml::Color*) zugeordnet werden. Darüber hinaus sind detaillierte Farbzugeordnungen (*fgml::AbstractTreeColorMapping*) möglich, z.B. um in RLP Farbkodierungen pro Entwicklungsphase zu definieren. Baumarten können zudem in Reduktionsgruppen für den Zuwachs (*fgml::IncrementReductionGroup*) zusammengefasst werden, denen wiederum Reduktionsfaktoren (*fgml::IncrementReduction*) abhängig von der Bestandesdichte zugeordnet werden kann.

Auf der anderen Seite werden Ertragstafeln (*fgml::AbstractYieldTable*) definiert. Jeder Baumart kann eine primäre Ertragstafel (*yieldTable*) sowie bei Bedarf mehrere mögliche alternative Tafeln (*alternativeYieldTables*) für spezielle Anwendungsfälle einer abweichenden Bonitierung zugeordnet werden. Letztere können dann konkreten Baumartenzeilen (*fgml::AbstractStandAttributeSet*) fallweise zugeordnet werden (vgl. Abschnitt 3.5). Ertragstafeln können dabei mit relativer (*fgml::AbstractYieldTableRelativeValuation*) oder absoluter (*fgml::AbstractYieldTableAbsoluteValuation*) Bonitierung definiert sein. Die Inhalte jeder Tafel werden über entsprechende Einträge (*fgml::AbstractYieldTableEntry*) definiert, die die entsprechenden Kenngrößen umfassen.

dies über eine eigene Klassenstruktur gesteuert, weil hier je nach Ermittlungsart noch zusätzliche Parameter gespeichert werden können. Dabei abgebildete Ansätze sind:

- *fgml::StockSamplePlots[Region/Enterprise]* – Stichprobenbasierte Vorratsschätzung [für Regionen/Betriebe]
- *fgml::StockYieldTable* – Ertragstafelbasierte Vorratsschätzung
- *fgml::StockAngleCountMethod* – Ertragstafelbasierte Vorratsschätzung mit Winkelzählprobe
- *fgml::AbstractStockFellingProduce* – Vorrat aus Maßnahmen
- *fgml::StockFelling* – Vorrat aus Hiebsergebnis
- *fgml::StockTending* – Vorrat aus Pflegemaßnahme
- *fgml::StockLogVolumeTable* – Massenfortschreibung
- *fgml::AbstractStockEstimate* – Vorratsschätzung
- *fgml::StockEstimateRest* – Vorratsschätzung (Restbestand)
- *fgml::StockEstimateStand* – Vorratsschätzung (Bestand)
- *fgml::StockEstimateStandAbsolute* – Vorratsschätzung (Bestand absolut)
- *fgml::AbstractStockRemoteSensing* – Vorratsschätzung durch Fernerkundung
- *fgml::StockRemoteSensingNDOM* – Fernerkundungs-/ertragstafelbasierte Vorratsschätzung
- *fgml::StockRemoteSensingPointCloud* – Schätzung über LIDAR-Punktwolke
- *fgml::StockCompleteEnumeration* – Vollklappung

4 Anhang: Spezifikation

Die vollständige Spezifikation von ForestGML finden Sie in kürze unter <https://www.kwh40.de/veroeffentlichungen/>.

5 Anhang: Nutzung von ForestGML in ForestML 4.0

Abbildung 5-1 zeigt einen beispielhaften DZ Wald (siehe Kapitel 1). *ForestGMLInventoryData* ist hier ein ForestML 4.0-Merkmal, das den detaillierten, nativen ForestGML-Datensatz enthält. Die anderen ForestML 4.0-Merkmale (*StandAttributeSet* mit *TreeSpecies*, *BasalArea*, *NumberOfStems* etc.) orientieren sich an den entsprechenden Modellierungsbausteinen aus ForestGML (siehe dieser Standpunkt) und beschreiben die grundlegenden Eigenschaften des Waldbestands bereits auf ForestML 4.0-Ebene.

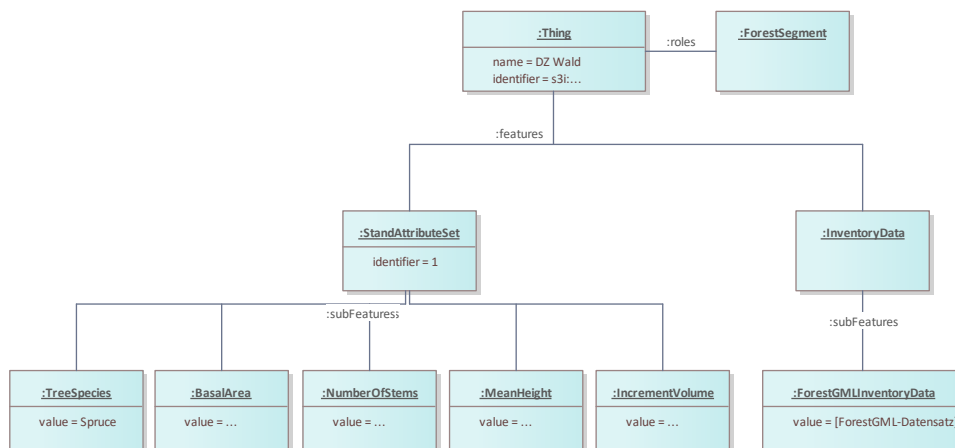


Abbildung 5-1: Integration in Wald und Holz 4.0: ForestML 4.0-basierter Digitaler Zwilling Wald mit Detail-Beschreibung im hier vorgestellten ForestGML-Format zur Waldinventur