



Geodateninfrastruktur
Bayern



Technische Universität München
Ingenieur fakultät Bau Geo Umwelt
Lehrstuhl für Geoinformatik
Univ.-Prof. Dr. rer. nat. Thomas H. Kolbe

Datenmodelltransformationswerkzeuge für Geodaten - ein Vergleich anhand ausgewählter Anwendungsfälle

Sebastian Georg Kroiß

Master's Thesis

Bearbeitung: 01.04.2015 - 30.09.2015
Studiengang: Geodäsie und Geoinformation (Master)
Betreuer: Dipl.-Inf. Tatjana Kutzner
Betreuer: Dr. Astrid Feichtner

2015

Zusammenfassung

In der Politik und zahlreichen weiteren Anwendungsfeldern werden kombinierbare Geodaten als Grundlage für eine effiziente Planung benötigt. Allerdings liegen Geodaten aus verschiedenen Quellen, insbesondere in Grenzregionen, häufig in heterogener und nur schwer kombinierbarer Fassung vor. Durch die Richtlinie INSPIRE des Europäischen Rates und der Kommission wurde eine Grundlage zur Gestaltung einer einheitlichen Geodateninfrastruktur in Europa geschaffen. Die Datenhalter von Geodaten wurden durch die Umsetzung der Richtlinie mit entsprechenden Gesetzen auf nationaler Ebene dazu verpflichtet, ihre Datensätze zu den betreffenden Geodaten-Themen fortan INSPIRE-konform bereitzustellen. Dazu müssen die bestehenden Datensätze aus ihren originären Datenmodellen in die neuen Datenmodelle von INSPIRE mit semantischen Datenmodelltransformationen überführt werden. Auch abseits von INSPIRE ist die semantische Datenmodelltransformation Gegenstand von aktuellen Forschungen in der Geoinformatik. Zur Durchführung von semantischen Datenmodelltransformationen stehen eine Reihe von Datenmodelltransformationswerkzeugen zur Verfügung. Diese Master's Thesis führt anhand der Transformation von drei ausgewählten Anwendungsfällen aus der Geodateninfrastruktur Bayerns (GDI-BY) nach INSPIRE eine Bewertung der drei ausgewählten Datenmodelltransformationswerkzeuge Feature Manipulation Engine (FME), GeoKettle und HUMBOLDT Alignment Editor (HALE) durch. Bei den ausgewählten Anwendungsfällen handelt es sich um Schutzgebietsdaten des Denkmal- und des Naturschutzes, sowie um das CityGML-Modell der Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (AdV). Die Regeln zur Durchführung der Transformationen wurden in Abbildungstabellen plattformunabhängig vorab zur Verfügung gestellt. Diese Regeln mussten auf die Transformationsansätze der Tools übertragen werden. Aus den Transformationsansätzen konnten Besonderheiten der Werkzeuge bei der Vorgehensweise aufgedeckt und auch Fehlerquellen identifiziert werden. Die internen Prozesse des Werkzeugs FME wurden eingehender untersucht und dabei wurde auch auf vorgenommene Paradigmenwechsel durch die Software von objektorientiertem zu relationalem Paradigma eingegangen. Zudem wurden die Grenzen der Darstellbarkeit von Abbildungsregeln zu Datenmodelltransformationen in Tabellen untersucht. Das Ziel der Arbeit ist die Evaluierung der Ergebnisse aus den Transformationen und die Bewertung der Tools hinsichtlich ihrer Eignung für die ausgewählten Anwendungsfälle. Zur Evaluierung und Bewertung der Werkzeuge wurde ein Bewertungsschema im Rahmen dieser Thesis entwickelt und angewandt. Aus den Erkenntnissen während der Durchführung der Transformationen und weiteren Überlegungen wurden zudem Grenzen in der Machbarkeit von semantischen Datenmodelltransformationen im Allgemeinen ausgemacht. Mit den Werkzeugen FME und HALE konnten in zwei der Anwendungsfälle korrekte Ergebnisse erzielt werden, womit diese Tools als geeignet eingestuft werden konnten. Im Anwendungsfall CityGML war HALE nicht in der Lage die Geometrien der Gebäudehülle zu transferieren. Das Tool GeoKettle stellte sich gänzlich als ungeeignet für die Transformation nach INSPIRE mit Transferformaten heraus, da entsprechende Funktionen fehlen und die Geometrie nicht gelesen oder bearbeitet werden konnte. Aus der Bewertung ging FME mit knappem Vorsprung vor HALE als die am Besten bewertete Software hervor.

Abstract

In politics and numerous other application fields, combined geographical data is required as a base for efficient planning. However, geodata available from various sources often only exists in heterogeneous formats which are difficult to combine in particular when using international data providers. INSPIRE, a directive created by the European Council and the European Commission, establishes a basis for a consistent infrastructure for spatial data within Europe. The national laws implementing this directive require the providers of geographical information in the European Union to provide data, relevant to the themes that INSPIRE applies to, in compliance with the formats required for exchange. Existing data sets must therefore be transformed from their original data models into INSPIRE's data models by performing semantic data model transformations. These semantic data model transformations are not only a topic treated within INSPIRE but also form a focus of current research.

A number of tools are available to perform semantic data model transformations. Within the scope of this thesis three tools were tested for their performance and accuracy including Feature Manipulation Engine (FME), GeoKettle and HUMBOLDT Alignment Editor (HALE). They were used to transform three selected origin data models from the Bavarian geodata infrastructure (GDI-BY) into three correspondent data models of INSPIRE. The case studies are data models on listed monuments, natural reserves and the CityGML model created by the Working Committee of the Surveying Authorities of the States of the Federal Republic of Germany (AdV).

Transformations are organized by mapping rules that were provided by the data holders in tables. These rules were used to define the mappings approaches of the data model transformation tools. The characteristics of each tool were investigated in search of sources of errors. In addition, the internal processes of FME were analyzed with respect to the change of paradigm from object oriented to relational. Furthermore, the mapping tables' capacities were analyzed regarding the representability of mapping rules for semantic data transformations in tables.

The aim of this thesis is to evaluate the results from the performed transformations and to evaluate the ability of the tools for the selected case studies. An evaluation procedure has been developed and applied to the tools. The findings obtained during the performed transformations are used to indicate the boundaries of the feasibility to perform semantic data model transformations.

Both FME and HALE returned satisfying results in two of the case studies, they are therefore considered as valide. HALE was not able to transfer the geometrical data of the building envelopes in the third case study (CityGML). GeoKettle proved unsuitable for transformations to INSPIRE as the necessary functions are missing and the geometry either can not be read or can not be altered. The comparison therefore rates FME as the most fitting software with a narrow lead on HALE.

Inhaltsverzeichnis

| | |
|--|-----------|
| Abbildungsverzeichnis | IX |
| Tabellenverzeichnis | X |
| 1 Motivation und Aufgabenstellung | 2 |
| 1.1 Motivation | 2 |
| 1.2 Aufgabenstellung | 4 |
| 1.3 Aufbau der Masterarbeit | 6 |
| 1.4 Verwandte Forschungsarbeiten | 7 |
| 2 Grundlagen der Datenmodellierung und Datenmodelltransformation | 8 |
| 2.1 Einführung in die Datenmodellierung | 8 |
| 2.1.1 Daten | 8 |
| 2.1.2 Datenmodell und Datenmodellierung | 9 |
| 2.1.3 Ebenen der Datenmodellierung | 11 |
| 2.1.4 Modellierungssprachen | 14 |
| 2.2 Relevante Normen, Spezifikationen und Vorschriften | 15 |
| 2.2.1 Interoperabilität | 17 |
| 2.2.2 INSPIRE | 17 |
| 2.2.3 Gesetzgebung auf Ebene der Mitgliedsstaaten | 19 |
| 2.3 Datenmodellierung für Geodaten | 20 |
| 2.3.1 Heterogenität von Geodaten | 22 |
| 2.3.2 Geodateninfrastrukturen | 23 |
| 2.3.3 Datentransfer | 23 |
| 2.4 Semantische Datenmodelltransformation und Abbildungssprachen | 24 |
| 2.4.1 Semantische Datenmodelltransformation | 24 |
| 2.4.2 Abbildungssprachen | 28 |
| 2.5 Objektorientiertes und Relationales Paradigma | 29 |
| 3 Beschreibungen der gewählten Fachanwendungen und der Software | 31 |
| 3.1 Originäre Datenmodelle | 31 |
| 3.1.1 Denkmalschutzgebiete in Bayern (Landesamt für Denkmalpflege) | 31 |
| 3.1.2 Schutzgebiete in Bayern (Landesamt für Umwelt) | 35 |
| 3.1.3 AAA 6.0 CityGML 1.0 | 38 |
| 3.1.4 Klassifizierung der ausgewählten Anwendungsfälle | 43 |

| | | |
|----------|--|-----------|
| 3.2 | INSPIRE-Datenmodelle | 43 |
| 3.2.1 | INSPIRE Data Specification on Protected Sites | 44 |
| 3.2.2 | INSPIRE Data Specification on Buildings | 45 |
| 3.3 | Beschreibungen der Transformationswerkzeuge | 47 |
| 3.3.1 | Feature Manipulation Engine | 47 |
| 3.3.2 | GeoKettle | 49 |
| 3.3.3 | HUMBOLDT Alignment Editor | 51 |
| 4 | Methodisches Vorgehen | 54 |
| 4.1 | Ausgangssituation | 54 |
| 4.2 | Analyse der Heterogenität zwischen den Datenmodellen | 55 |
| 4.2.1 | Heterogenität der korrespondierenden Elemente | 55 |
| 4.2.2 | Heterogenität im Umfang der Modelle | 55 |
| 4.2.3 | Geometrische Heterogenität | 57 |
| 4.3 | Gestaltung von Abbildungstabellen | 57 |
| 4.4 | Lösungsansatz der Transformationsaufgabe | 59 |
| 4.5 | Erstellung eines Bewertungsschemas | 60 |
| 5 | Darstellung der Transformationsergebnisse | 62 |
| 5.1 | Analyse der Abbildungstabellen | 62 |
| 5.1.1 | Analyse der Abbildungstabelle Denkmalschutzgebiete - Protected Sites | 62 |
| 5.1.2 | Analyse der Abbildungstabelle Naturschutzgebiete - Protected Sites | 63 |
| 5.1.3 | Analyse der Mapping-Tabelle AAA6.0 CityGML1.0 - INSPIRE BuildingsCore3DLoD2 | 64 |
| 5.2 | Ergebnisse für Feature Manipulation Engine | 65 |
| 5.2.1 | Transformationsansatz von FME | 65 |
| 5.2.2 | Transformation der Geometrie mit FME | 66 |
| 5.2.3 | Transformation der Semantik mit FME | 66 |
| 5.2.4 | Lösungsstrategien für die klassifizierten semantischen Relationen | 68 |
| 5.2.5 | Paradigmenwechsel während der Transformation mit FME | 69 |
| 5.2.6 | Performance von FME | 71 |
| 5.2.7 | Berichtswesen und Validierung in FME | 73 |
| 5.3 | Ergebnisse für GeoKettle | 73 |
| 5.3.1 | Transformationsansatz von GeoKettle | 73 |
| 5.3.2 | Import von Daten | 74 |
| 5.3.3 | Transformation der Geometrie mit GeoKettle | 74 |
| 5.3.4 | Transformation der Semantik mit GeoKettle | 76 |
| 5.3.5 | Ergebnis der Transformationen und Berichtswesen | 76 |
| 5.4 | Ergebnisse für HUMBOLDT Alignment Editor | 77 |
| 5.4.1 | Transformationsansatz von HALE | 77 |
| 5.4.2 | Transformation der Geometrie mit HALE | 79 |

| | | |
|----------|--|------------|
| 5.4.3 | Transformation der Semantik mit HALE | 80 |
| 5.4.4 | Lösungsstrategien für die klassifizierten semantischen Relationen | 82 |
| 5.4.5 | Performance von HALE | 83 |
| 5.4.6 | Verwendung von HALE abseits von INSPIRE | 83 |
| 5.4.7 | Berichtswesen und Validierung in HALE | 83 |
| 5.5 | Vergleich der Ergebnisse | 84 |
| 6 | Bewertung und Vergleich der Transformationswerkzeuge | 85 |
| 6.1 | Problemstellungen | 85 |
| 6.1.1 | Eigenheiten der Datenmodelltransformatiionswerkzeuge | 85 |
| 6.1.2 | Grenzen der Datenmodelltransformation | 86 |
| 6.1.3 | Grenzen der Darstellung in Abbildungstabellen | 89 |
| 6.2 | Festlegung der Bewertungskriterien | 91 |
| 6.2.1 | Kriterien zur Evaluierung der Ergebnisse | 91 |
| 6.2.2 | Kriterien zur Beurteilung der Datenmodelltransformatiionswerkzeuge | 93 |
| 6.3 | Darstellung und Beschreibung des Bewertungsschemas | 102 |
| 6.4 | Bewertung von Feature Manipulation Engine | 104 |
| 6.4.1 | Bewertung von FME nach Bewertungskatalog A | 105 |
| 6.4.2 | Bewertung von FME nach Bewertungskatalog B | 106 |
| 6.4.3 | Bewertung von FME nach Bewertungskatalog C | 111 |
| 6.4.4 | Abschließende Bewertung von FME | 113 |
| 6.5 | Bewertung von GeoKettle | 113 |
| 6.5.1 | Bewertung von GeoKettle nach Bewertungskatalog A | 113 |
| 6.5.2 | Bewertung von GeoKettle nach Bewertungskatalog B | 114 |
| 6.5.3 | Bewertung von GeoKettle nach Bewertungskatalog C | 118 |
| 6.5.4 | Abschließende Bewertung von GeoKettle | 119 |
| 6.6 | Bewertung von HUMBOLDT Alignment Editor | 120 |
| 6.6.1 | Bewertung von HALE nach Bewertungskatalog A | 120 |
| 6.6.2 | Bewertung von HALE nach Bewertungskatalog B | 121 |
| 6.6.3 | Bewertung von HALE nach Bewertungskatalog C | 126 |
| 6.6.4 | Abschließende Bewertung von HALE | 128 |
| 6.7 | Vergleichende Wertung | 128 |
| 7 | Schlussfolgerung und Ausblick | 130 |
| 7.1 | Zusammenfassung der Ergebnisse und Schluss | 130 |
| 7.2 | Fazit und Empfehlungen | 131 |
| 7.3 | Ausblick auf weitere Fragestellungen | 131 |
| | Literaturverzeichnis | 133 |
| | Anhänge | 139 |

| | | |
|----------|--|------------|
| A | Abkürzungsverzeichnis | 140 |
| B | Digitaler Anhang (DVD) | 142 |
| C | Klassendiagramme der INSPIRE Data Specifications | 143 |
| C.1 | INSPIRE Data Specification on Protected Sites: Objektklasse und Codelists mit deutscher Erweiterung | 144 |
| C.2 | «featureType» GermanMonument und «codeList» StateLegalDefinitionDesignationRecordValue | 145 |
| C.3 | INSPIRE Data Specification on Buildings: Buildings Core 3D | 146 |
| D | Klassifikation von Transformationsansätzen | 147 |

Abbildungsverzeichnis

| | | |
|------|--|----|
| 2.1 | Bsp. Klasse «Gebäude» | 8 |
| 2.2 | Modellierung eines Realweltausschnitts in einem Datenmodell | 11 |
| 2.3 | Ebenen der Datenmodellierung: MDA und Datenbanken | 12 |
| 2.4 | Hierarchie der Modellierungssprachen bei der Erzeugung von GML-Instanzen | 15 |
| 2.5 | Systematische Übersicht über einschlägige Normen und Standards | 16 |
| 2.6 | Rahmenwerk zur Entwicklung der INSPIRE <i>Data Specifications</i> | 19 |
| 2.7 | Die Geodaten-Themen der Anhänge I bis III von INSPIRE | 20 |
| 2.8 | Semantische Nutzeranforderungen an CityGML | 22 |
| 2.9 | Aufbau der GDI-BY | 23 |
| 2.10 | Prinzip der Semantischen Datenmodelltransformation mit ETL-Tools | 25 |
| 2.11 | Prozess der Semantischen Datenmodelltransformation | 28 |
| 3.1 | FeatureType Baudenkmaeler: Attribute und Anmerkungen | 33 |
| 3.2 | FeatureType Bauensembles: Attribute und Anmerkungen | 33 |
| 3.3 | FeatureType Bodendenkmaeler: Attribute und Anmerkungen | 33 |
| 3.4 | Geometrie der FeatureTypes der Denkmaldaten des BLfD | 34 |
| 3.5 | Darstellung der Denkmalarten des BLfD | 34 |
| 3.6 | FeatureType biosphaerenreservat: Attribute und Anmerkungen | 36 |
| 3.7 | FeatureTypes fauna_flora_habitat_gebiet und vogelschutzgebiet: Attribute und Anmerkungen | 36 |
| 3.8 | FeatureTypes landschaftsschutzgebiet, nationalpark, naturpark und naturschutzgebiet: Attribute und Anmerkungen | 37 |
| 3.9 | Darstellung der Schutzgebietsdaten des LfU am Beispiel von Naturparks | 37 |
| 3.10 | Generalisierung von Gebäudemodellen in CityGML (LoD) | 38 |
| 3.11 | CityGML-Profil für LoD2 (Building) der AdV (2014) | 41 |
| 3.12 | Darstellung des Datensatzes in 3D-Ansicht [Popp 2015, S.14] und als Draufsicht | 42 |
| 3.13 | Darstellung der Transformationsfälle | 44 |
| 3.14 | Inhalt und Struktur der Anwendungsschemata für INSPIRE DS BU | 46 |
| 3.15 | FME: Übersicht über Benutzeroberfläche | 48 |
| 3.16 | GeoKettle: Übersicht über Benutzeroberfläche | 51 |
| 3.17 | HALE: Übersicht Benutzeroberfläche | 53 |
| 3.18 | HALE panel - <i>Schema Explorer</i> | 53 |
| 3.19 | HALE panel - <i>Alingment Editor</i> | 53 |

| | | |
|------|---|-----|
| 4.1 | Klassifikation der semantischen Relationen und Heterogenitäten nach Fichtinger (2011) | 56 |
| 4.2 | Template für Abbildungstabellen zum Schema-Mapping | 58 |
| 4.3 | Beispiel für ein Schema-Mapping in HALE mit Transformationsfall II | 59 |
| 5.1 | FME: Beispiel für die Ausgabe der Geometrie in den Transformationsfällen I & II . . . | 67 |
| 5.2 | FME: Interne Handhabung von Attributen (links) und FME Geometriemodell (rechts) (c) Safe Software Inc. | 70 |
| 5.3 | FME: Zwischenspeicherung von Elementstrukturen mit XFMLaps | 70 |
| 5.4 | FME: Elemente zur relationalen Strukturierung | 71 |
| 5.5 | FME: Repräsentation der Attribute zur Laufzeit | 71 |
| 5.6 | GeoKettle: Fehlermeldung beim Einlesen der Daten des LfU | 74 |
| 5.7 | GeoKettle: Fehlermeldung bei der Abfrage eines WFS | 75 |
| 5.8 | HALE: Beispiel für die Ausgabe der Geometrie in den Transformationsfällen I & II . . . | 79 |
| 5.9 | HALE: Anzeige von Attributen und deren Eigenschaften | 81 |
| 5.10 | HALE: Erzeugung eines Attributwertes mit Groovy Script | 82 |
| 6.1 | Beispiel zweier Elemente des grafischen Verzweigungsdiagramms | 104 |

Tabellenverzeichnis

| | | |
|------|--|-----|
| 3.1 | Überblick über den Netzdienst des Bayerischen Landesamtes für Denkmalpflege (BLfD) | 32 |
| 3.2 | Überblick über den Netzdienst des Bayerischen Landesamtes für Umwelt (LfU) | 35 |
| 3.3 | Attribute des verwendeten Datensatzes für CityGML | 42 |
| 3.4 | Klassifizierung der Anwendungsfälle | 43 |
| 3.5 | Liste der in INSPIRE DS PS (2014) verwendeten Typen | 45 |
| 3.6 | Übersicht über verwendete Software für semantische Transformationen | 47 |
| 3.7 | FME Desktop, Datentabelle der genutzten Edition [Safe Inc. 2015] | 48 |
| 3.8 | GeoKettle, Datentabelle der genutzten Edition [Spatialytics 2015] | 50 |
| 3.9 | HALE, Datentabelle der genutzten Edition [ESDIN 2015] | 52 |
| 4.1 | Benennung der Abbildungstabellen | 58 |
| 5.1 | FME: Kennwerte zur Performance | 72 |
| 5.2 | HALE: Funktionsbibliothek | 78 |
| 6.1 | Klassifizierung der Transformationsansätze | 87 |
| 6.2 | Beispiel zur tabellarischen Darstellung für ein Bewertungskriterium | 103 |
| 6.3 | FME: Bewertung der Software nach Bewertungskatalog A: Evaluierung der Ergebnisse | 106 |
| 6.4 | FME: Bewertung der Software nach Bewertungskatalog B: Leistungsvermögen der Software | 111 |
| 6.5 | FME: Bewertung der Software nach Bewertungskatalog C: Sonstige Kategorien | 112 |
| 6.6 | GeoKettle: Bewertung der Software nach Bewertungskatalog A: Evaluierung der Ergebnisse | 114 |
| 6.7 | GeoKettle: Bewertung der Software nach Bewertungskatalog B: Leistungsvermögen der Software | 118 |
| 6.8 | GeoKettle: Bewertung der Software nach Bewertungskatalog C: Sonstige Kategorien | 119 |
| 6.9 | FME: Bewertung der Software nach Bewertungskatalog A: Evaluierung der Ergebnisse | 121 |
| 6.10 | HALE: Bewertung der Software nach Bewertungskatalog B: Leistungsvermögen der Software | 126 |
| 6.11 | HALE: Bewertung der Software nach Bewertungskatalog C: Sonstige Kategorien | 127 |
| 6.12 | Punktwertung der ETL-Tools im Vergleich | 128 |

Kapitel 1

Motivation und Aufgabenstellung

1.1 Motivation

In zahlreichen Politikfeldern, sowie auch in der freien Wirtschaft, leisten Geodaten einen wichtigen und mittlerweile unentbehrlichen Beitrag zu Prozessen der Entscheidungsfindung, Meinungs- und Maßnahmenbildung. Somit stellen sie eine Informationsgrundlage dar, die gestützt auf Analyse der Daten, eine fundierte und objektive Definition von Anforderungen, Kriterien und Zielstellungen in Prozessen ermöglicht: Geodaten generieren Planungskompetenz für das eGovernment¹.

Für die Allgemeinheit werden Geodaten allerdings erst dann verständlich lesbar und nutzbar, wenn ihnen ein öffentlich zugängliches Datenmodell zugrunde liegt, welches ihre Bedeutung und Struktur (bezüglich Ausdrücken und Anordnung) beschreibt. Die Gestaltung der Datenmodelle hängt dabei in der Regel davon ab, wer der Urheber der Daten ist bzw. in wessen Interesse die Datenerhebung vonstattengeht. Unterschiede in den Modellen können beispielsweise in den unterschiedlichen Plattformen begründet sein, andererseits aber auch in der regionalen Gebietsbegrenzung, innerhalb derer ein Datenmodell zur Anwendung kommt.

Eine solche regionsspezifische Abgrenzung stellen für gewöhnlich Ländergrenzen dar. Überschreitet man die Staatsgrenzen in der Europäischen Union (EU), beispielsweise über ein Grenzgewässer hinweg, so ändert sich auch die Beschreibung eines Objektes (z.B. die Beschreibung des Grenzgewässers) bzw. die Definition von Datenmodellen diesbezüglich, obwohl dieses erklärtermaßen im (u.a) umweltpolitischen Interessenfeld beider angrenzender Staaten liegt. Demzufolge ist es auch unumgänglich, in der Durchführung von Planungsaufgaben über hoheitliche Grenzen hinweg, einen Mehraufwand bei der Verarbeitung von heterogenen Datenquellen bzw. -modellen in Kauf nehmen zu müssen. Zudem verlagern sich die Entscheidungsprozesse zunehmend von dezentralen - bzw. aus europäischer Perspektive regionalen - Entscheidungsträgern hin zu einer zentralisierten Entscheidungsebene: Der Europäischen Union.

In diesem Sinn nimmt beispielsweise die EU ihre Aufgabe als überstaatliche Organisation wahr, gemäß dem Subsidiaritätsprinzip (EU-Vertrag 2007, Art.3b (3)) für alle Mitgliedsstaaten geltende Vorschriften und Maßnahmen im Bereich der Umweltpolitik zu erlassen. Sowie auch, um Aktionsprogramme zu beschließen [EU Charta 2010, Art.192] und somit eine Politik auf Basis gemeinschaftlicher Interessen- und Willensbildung zu ermöglichen. Diese Zentralisierung von staatlichen Aufgaben

¹Das eGovernment ist eine Initiative des Freistaates Bayern zur Förderung der Nutzung von Geodaten in der Politik.

geht auch mit einer steigenden Komplexität von Entscheidungsprozessen einher, womit wiederum die Anforderungen an die Verarbeitbarkeit, Aussagekraft und Quantität bzw. Qualität von Daten als Informationsbasis steigen.

Um die Grenzen abzubauen, welche die Heterogenität der digitalisierten Welt den Anwendern setzt, haben sich eine Vielzahl an Gremien, Vereinen und Organisationen zusammengefunden (z.B. OGC²). Diese haben es sich zur Aufgabe gesetzt, neue Standards maßgeblich zu gestalten und relevante bestehende Standards (u.a. in den Bereichen der Datenhaltung, -übertragung, -verarbeitung und -modellierung) sinnvoll zu vereinheitlichen. Somit soll eine Interoperabilität im Sinne der Fähigkeit zur Zusammenarbeit heterogener Systeme [Kutzner und Eisenhut 2010, S.2] ermöglicht werden.

Im März 2007 wurde von der EU zu diesem Zweck eine Richtlinie (INSPIRE-Richtlinie) verabschiedet, mit dem Ziel eine europaweit konsistente und einheitlich modellierte Infrastruktur für Geodaten zu schaffen, zum Nutzen einer gemeinschaftlichen Umweltpolitik der Mitgliedstaaten [Europäisches Parlament und Rat 2007]. Mit dieser gemeinschaftlichen Geodatenpolitik in Europa sollen „(...) die Probleme bei der Verfügbarkeit, Qualität, Organisation, Zugänglichkeit und gemeinsamen Nutzung von Geodaten (...)“ [Europäisches Parlament und Rat 2007; S.1 (3)] behoben werden, die bislang eine effektive Planung über Ländergrenzen hinweg erschwerten. Insbesondere wird das Ziel verfolgt, den Austausch und die Nutzung von Geodaten zwischen Ländern und Plattformen stark zu vereinfachen und mit dieser Interoperabilität „(...) einen signifikanten Mehrwert für andere Gemeinschaftsinitiativen (...)“ [Europäisches Parlament und Rat 2007; S.2 (10)], wie z.B. Galileo³ und GMES⁴, zu schaffen.

Die Umsetzung der Richtlinie zum Aufbau einer Europäischen Geodateninfrastruktur (EGDI) bedeutet in der Folge auch, dass die vorliegenden heterogenen Konzepte zu Datenmodellen zu dem oben erwähnten interoperablen Konzept der INSPIRE-Richtlinie vereinheitlicht werden müssen. Die Datenhalter relevanter Konzepte in der EU stehen somit vor der Aufgabe ihre ausgestalteten Datenmodelle in die neuen zu INSPIRE konformen Konzepte zu überführen. Nun stellt sich die Frage, wie und mit welchen Mitteln diese Überführung zu bewerkstelligen ist. Über den Kontext von INSPIRE hinaus erlangt die Zusammenführung von heterogenen Konzepten im Bereich der Geoinformatik⁵ generell zunehmend an Bedeutung. Insbesondere vor dem Hintergrund der fortschreitenden digitalen Erschließung der Welt, da in immer stärkerem Ausmaß rein technisch fundierte (und zum Teil bislang autark vorgehaltene) Datenmodelle nach Georeferenzierung und Verknüpfung mit Objekten der realen Welt verlangen (z.B.: Automatisierung in der Fahrzeugtechnik; Konzepte der Indoor-Navigation u.v.m.).

Die vorliegende Master's Thesis, die in Kooperation durch die Technische Universität München und das Bayerische Landesamt für Digitalisierung, Breitband und Vermessung ausgeschrieben wurde, befasst sich mit der Transformation zwischen Datenmodellen mit besonderem Fokus auf die Transformation hin zu INSPIRE-konformen Modellen.

²Open Geospatial Consortium (OGC): <http://www.opengeospatial.org/> - abgerufen am 23.07.2015

³Galileo (Europäisches Satellitensystem zur Zeitgebung und Navigation): http://ec.europa.eu/enterprise/policies/satnav/index_de.htm - abgerufen am 23.07.2015

⁴Global Monitoring for Environment and Security (GMES): <http://www.d-gmes.de/> - abgerufen am 23.07.2015

⁵Die Geoinformatik ist ein Teilbereich der Informatik mit Spezialisierung auf räumliche Daten [Wikipedia 2015a].

1.2 Aufgabenstellung

Datenmodelle liegen - wie bereits zuvor erwähnt - in vielgestaltigen Ausprägungen bezüglich ihrer inhaltlichen und strukturellen Konzepte vor, obwohl sie möglicherweise ein- und denselben thematischen Hintergrund beschreiben. Beispielsweise ergibt sich hinsichtlich der Datenbereitstellung im Hinblick auf INSPIRE die Problematik, dass die originären Datenmodelle für Geodaten in Deutschland nicht konform sind mit den angestrebten Datenmodellen, den sogenannten INSPIRE *Data Specifications*, die zur Umsetzung der INSPIRE-Richtlinie im europaweiten Kontext gestaltet wurden (z.B. [INSPIRE DS PS 2014]).

Folglich ist es notwendig, die bestehenden originären Datenmodelle (z.B. Naturschutzgebiete) in die konform zur INSPIRE-Richtlinie neu gestalteten Datenmodelle mittels semantischen Datenmodelltransformationen (siehe Abschnitt 2.4) zu überführen. Hieraus ergibt sich Punkt (1):

1) Im Rahmen dieser Master's Thesis sollen ausgewählte Datenmodelltransformationswerkzeuge (ETL⁶-Tools) für Geodaten anhand ausgewählter Anwendungsfälle verglichen werden.

Seitens des Bayerischen Landesamtes für Digitalisierung, Breitband und Vermessung (LDBV)⁷ wurden zwei exemplarische Anwendungsfälle zur Durchführung des praktischen Teils dieser Arbeit ausgewählt:

- a) Denkmalschutzgebiete des Bayerischen Landesamtes für Denkmalpflege und
- b) Naturschutzgebiete des Bayerischen Landesamtes für Umwelt.

Der dritte Anwendungsfall wurde von der TU München ausgewählt und betrifft:

- c) CityGML-Profil AAA⁸ 6.0 der Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (AdV).

Diese originären Anwendungsschemata (bzw. Quellschemata) sollen in die jeweiligen zu INSPIRE konformen Zielschemata transformiert werden. Bei den Zielschemata handelt es sich bezüglich (a) und (b) um: d) INSPIRE *Data Specification on Protected Sites* [INSPIRE DS PS 2014].

Bezüglich (c) ist das Zielschema e) INSPIRE *Data Specification on Buildings* [INSPIRE DS BU 2013].

Zur Durchführung der Transformationen wurden drei ETL-Tools ausgewählt:

- f) Feature Manipulation Engine (FME) von Safe Software Inc.,
- g) GeoKettle von Spatialytics und
- h) HUMBOLDT Alignment Editor (HALE), der im Rahmen des HUMBOLDT Framework entwickelt wurde.

Zu allen drei in dieser Master's Thesis relevanten Anwendungsfällen wurden von der TU München

⁶Extract - Transform - Load

⁷Landesamt für Digitalisierung, Breitband und Vermessung (LDBV):

<http://vermessung.bayern.de/service/organisation/lvg.html> - abgerufen am 23.07.2015

⁸AFIS - ALKIS - ATKIS (AAA): Modell der AdV zur Digitalisierung von Vermessungs- und Katasterdaten

und dem LDBV Abbildungstabellen (siehe Abschnitt 4.3) zur Verfügung gestellt. In diesen Tabellen wurden vorab plattformunabhängig die benötigten Konstrukte und Relationen zur konkreten Ausführung der semantischen Datenmodelltransformationen zwischen jeweiligem Quellschema (a) bis (c) und Zielschema (d) bzw. (e) exemplarisch festgelegt. In den Abbildungstabellen werden somit die semantischen Datenmodelltransformationen in Form eines sogenannten Mapping-Schemas dargestellt. Der semantischen Datenmodelltransformation als solcher und den Abbildungstabellen im Besonderen sind jedoch Grenzen gesetzt, welche es im Ansatz zu ergründen gilt. Dies führt zu den Punkten (2) und (3):

2) Im Rahmen dieser Master's Thesis soll erörtert werden, unter welchen Voraussetzungen der Darstellung von Abbildungen in Tabellen Grenzen gesetzt sind.

(3) Im Rahmen dieser Master's Thesis sollen die Grenzen der semantischen Datenmodelltransformation aufgezeigt werden. Dabei sind generell die Bedingungen zu hinterfragen, unter denen eine Datenmodelltransformation funktioniert und Voraussetzungen festzustellen, die geschaffen werden müssten, um ein Versagen der Transformationen zu verhindern.

Der Vorgang der semantischen Datenmodelltransformation, wie er in den ETL-Tools auf Basis von Transformationssprachen (siehe Abschnitt 2.4.2) implementiert ist, erfordert unter Umständen auch einen Paradigmenwechsel von objektorientiertem Paradigma hin zu relationalem Paradigma.

Dies ergibt Punkt (4) der Aufgabenstellung:

4) Im Rahmen dieser Master's Thesis soll geprüft werden, wie ein eventuell in Erscheinung tretender Übergang von objektorientiertem Paradigma zu einer programminternen relationalen Strukturierung beim Einlesen von Daten bzw. Anwendungsschemata und wieder zurück beim Schreiben der Zieldaten funktioniert. Es ist auch festzustellen, ob dann in den Zieldaten die korrekte objektorientierte Struktur wieder ausgegeben wird.

Sollte es im Rahmen dieses Paradigmenwechsels zu Fehlern kommen, oder sollten dadurch Fehlerquellen gegeben sein, sind diese zu erörtern.

1.3 Aufbau der Masterarbeit

In diesem **Kapitel 1** wurde bereits eine kleine Einführung in die Thematik gegeben. Darin wurde auch die Grundproblematik dargelegt, welche die Motivation zur Ausschreibung des Themas begründet (1.1). Anschließend wurde die sich hieraus ergebende konkrete Aufgabenstellung dieser Master's Thesis formuliert (1.2) und in kurzen Punkten zusammengefasst. Abschnitt 1.4 gibt einen Überblick über verwandte Forschungsarbeiten (related work).

In **Kapitel 2** soll ein grundlegendes Verständnis für den Fachbereich vermittelt werden (2.1). Von Bedeutung sind einige relevante Normen bzw. Vorschriften und Datenspezifikationen, auf denen u.a. die Gestaltung der ausgewählten Anwendungsfälle beruht (2.2). Nachfolgend werden die theoretischen Grundlagen der Datenmodellierung (2.3) und der semantischen Datenmodelltransformation (2.4), mit Fokus auf die Anwendung mit Geodaten, erläutert. Den Grenzen, die der semantischen Datenmodelltransformation und ihrer exemplarischen Darstellung gesetzt sind, widmet sich der nächste Abschnitt (2.5), während im Anschluss daran (2.6) mögliche Sprachparadigmen diskutiert werden.

In **Kapitel 3** erfolgt die Beschreibung und Analyse der gewählten Fachanwendungen der Quell- und Zielschemata (3.1 und 3.2). Anschließend werden im letzten Abschnitt des Kapitels (3.3) die zur Anwendung kommenden Datenmodelltransformationswerkzeuge im Detail vorgestellt.

Mit **Kapitel 4** beginnt die schriftliche Niederlegung des praktischen Teils der Master's Thesis. Zunächst wird die Ausgangssituation nochmals kurz zusammengefasst (4.1) bevor dann mögliche Quellen von Heterogenitäten zwischen Datenmodellen diskutiert werden (4.2). Nachfolgend wird das Prinzip der verwendeten Abbildungstabellen beschrieben und daraus resultierende Lösungsansatz als Grundlage der in den ETL-Tools gestalteten Transformationsregeln erörtert (4.3). In Abschnitt 4.5 wird dargelegt, wie beim Entwurf eines Bewertungsschemas zur Beurteilung der ETL-Tools vorgegangen wurde.

In **Kapitel 5** erfolgt zunächst die Analyse der Abbildungstabellen und daraus die Extraktion von Anforderungsprofilen der Anwendungsfälle an die Tools (5.1). Die Ergebnisse der in der Praxis durchgeführten semantischen Datenmodelltransformationen werden nachfolgend dargelegt (5.2 bis 5.4) und verglichen (5.5). Dabei wird näher auf die Vorgehensweisen eingegangen und es werden Besonderheiten der Tools aufgedeckt.

Auf Basis der in den vorherigen Kapiteln festgehaltenen Ergebnisse werden dann in **Kapitel 6** die Problemstellungen untersucht und dabei auch die Grenzen von Datenmodelltransformationen und Abbildungstabellen aufgedeckt (6.1). Darauf folgend werden Kriterien zu Bewertungskatalogen zusammengefasst, beschrieben und aufbereitet (6.2 bis 6.3) und zur Bewertung der Tools herangezogen (6.4 bis 6.6), wodurch eine abschließende Aussage über die Qualifizierung der ausgewählten Tools getroffen werden kann (6.7).

Zum Abschluss der Master's Thesis wird in **Kapitel 7** eine Zusammenfassung und Schlussfolgerung der resultierenden Erkenntnisse vorgenommen sowie ein Ausblick auf weitere Fragestellungen gewährt.

1.4 Verwandte Forschungsarbeiten

Es wurden bereits einige Forschungsarbeiten veröffentlicht, die eine ähnliche Thematik wie die vorliegende Arbeit behandeln. In der Diplomarbeit von Unger (2012) wurde mit dem ETL-Tool HUMBOLDT Alignment Editor eine semantische Datenmodelltransformation von österreichischen Katasterdaten zu INSPIRE DS on Cadastral Parcels durchgeführt, wobei sich das Tool als geeignet herausgestellt hat.

Durch Popp (2015) wurde mit FME eine semantische Datenmodelltransformation von CityGML in die auch hier relevante Datenspezifikation INSPIRE DS on Buildings [INSPIRE DS BU 2013] vorgenommen. Aus diesem Grund wurde der entsprechende Anwendungsfall mit FME in der vorliegenden Arbeit nicht mehr transformiert. Auch in dieser Arbeit ist die Transformation geglückt. Außerdem wurde durch Popp die auch in der vorliegenden Arbeit verwendete Mapping-Tabelle zum Anwendungsfall untersucht. Dabei wurde festgestellt, dass sich die Tabelle unter Einschränkungen als Grundlage für Abbildungsregeln nutzen lässt. Die Erkenntnisse aus der Bachelor's Thesis von Popp wurden zudem in dieser Arbeit berücksichtigt.

Horák et al. (2011) testeten das Tool HALE bei der Transformation von hydrographischen Daten nach INSPIRE und auch hier eine ausreichende Eignung der Software bescheinigen.

In Fichtinger (2011) wurde eine semantische Datenmodelltransformation von topographischen Geobasisdaten aus der Grenzregion Bodensee nach INSPIRE erfolgreich durchgeführt. Auch in dieser Arbeit wurde das Tool HALE angewendet, wobei es sich wiederum als geeignet empfahl.

Kapitel 2

Grundlagen der Datenmodellierung und Datenmodelltransformation

In diesem Kapitel wird erläutert, was unter den Begriffen Daten, Datenmodelle und Datenmodelltransformation zu verstehen ist und wie und auf welchen Grundlagen sich diese definieren lassen. Es wird auch eine Reihe von weiteren Begriffen und Konzepten vorgestellt, die in diesem Kontext von Bedeutung sind, insbesondere im Zusammenhang mit der Geoinformatik.

2.1 Einführung in die Datenmodellierung

2.1.1 Daten

Daten werden genutzt, um einen Wert - oder allgemeiner eine Aussage zu einem Objekt - in Form von Datenformaten (z.B. String für Text, Integer für Zahlen) zu speichern und auf Abruf wiederzugeben bzw. weiter zu verarbeiten. Abstrahiert betrachtet sind Daten die digitale¹ und somit auf Basis von Formalismen maschineninterpretierbare² Repräsentation von realen Objekten [Staub 2009, S.7]. Unter einem „realen Objekt“ kann man einerseits verstehen, dass es sich um ein in der Welt physisch präsent Objekt handelt, oder aber auch um dessen nicht physisch fassbare Eigenschaften. Darüber hinaus sind auch Daten existent, die fiktive Konstrukte ganz ohne physische Manifestation beschreiben (z.B. in der Finanzbranche), an dieser Stelle sind sie aber nicht von Bedeutung.

| Haus |
|---|
| + Geometrie: Geometrietyp + Baujahr: Datum |

Abb. 2.1 – Bsp. Klasse «Gebäude»

Betrachtet man als Beispiel für ein reales Objekt ein Haus, das durch ein stark vereinfachtes Datenmodell als simple Objektklasse mit Attributen (Eigenschaften) beschrieben werden soll (Abbildung 2.1), ist dessen signifikanteste Eigenschaft zunächst seine Form (d.h. Geometrie), welche eine als zuvor physisch fassbar beschriebene Eigenschaft darstellt. Ist nur die Grundfläche des Gebäudes von Be-

lang (z.B. zur Beschreibung der Liegenschaft), genügt der Gebäudegrundriss in Form einer zweidimensionalen Fläche (z.B. Polygon). Wird darüber hinaus eine dreidimensionale Repräsentation des

¹Daten können auch analog vorliegen.

²Maschineninterpretierbar bedeutet: Ein Text oder Code kann von einem Computer bzw. von Programmen gelesen und sinnvoll verarbeitet werden.

Gebäudes benötigt, müssen auch die weiteren baulichen Bestandteile wie z.B. Wände, Dächer oder eventuell auch kleinere Anbauten bzw. bauliche Ausprägungen miteinbezogen werden. Als weitere Eigenschaft könnte das Baujahr des Hauses verzeichnet werden. Da man dieses dem Haus nicht konkret ablesen kann, handelt es sich um eine nicht physisch fassbare Eigenschaft, die aber dennoch für diverse Anwendungen von Bedeutung ist. So könnte man aus dem Baujahr des Hauses auf den Zustand der Bausubstanz interpretierend rückschließen. Möchte man den Interpretationsspielraum konkretisieren, müsste man die Objektklasse um weitere beschreibende Attribute erweitern. Das Ausmaß der Detaillierung hängt dabei letztlich von den Wünschen des Nutzers und dem Zweck der Anwendung ab und kann eine große Spanne umfassen.

2.1.2 Datenmodell und Datenmodellierung

Den ersten Schritt bevor die Daten sozusagen „ins Leben gerufen“ werden können, stellt die Erstellung eines Datenmodelles dar. Unter dem Begriff der Datenmodellierung versteht man dementsprechend den Vorgang/Prozess, ein Datenmodell festzulegen und definierend bzw. beschreibend niederzuschreiben. Datenmodelle legen eine Struktur fest, um die Form und den Inhalt von Daten zu beschreiben und damit dem Nutzer erkenntlich zu machen, was beschrieben wurde und wie es beschrieben wurde. Die äußere Form eines Datenmodells wird durch die Syntax wiedergegeben.

Begriffsbestimmung

Syntax: In den Sprachwissenschaften beschreibt die Syntax die strukturell geordnete und auf Regeln (Formalismen) basierende Verknüpfung von sprachlichen Elementen (Ausdrücke) zu einer reproduzierbaren und einen Sinn bildenden Kombination [Duden 2015].

Datenmodelle bedienen sich also eines sprachlichen Formalismus, mit dessen Hilfe der Inhalt der Daten in definierter Form wiedergegeben wird. Die sprachlichen Formalismen setzen sich zu einer sogenannten Modellierungssprache zusammen, die ihrerseits wiederum gewissermaßen auf Formalismen beruht. Auf die Modellierungssprachen wird in Abschnitt 2.1.2 näher eingegangen.

Der Inhalt eines Datenmodells, also „was“ dadurch beschrieben werden soll, wird durch dessen Semantik wiedergegeben.

Begriffsbestimmung

Semantik: Unter dem Begriff Semantik wird die Bedeutung bzw. der Inhalt eines Ausdrucks, Konstruktes oder Objektes verstanden [Kutzner und Eisenhut 2010, S.13; Duden 2015]. Die Semantik steht immer auch im Zusammenhang mit der Syntax, da syntaktisch falsch zusammengesetzte Strukturen keine Semantik zugewiesen werden kann.

Im vorherigen Beispiel wurde ein Gebäude anhand von dessen Geometrie und Baujahr als Objekt mit Attributen dargestellt. Semantisch betrachtet beschreibt das Attribut der Geometrie die Form des Gebäudes (z.B. in Form von Koordinatenlisten), während das Baujahr semantisch gesehen u.a. das Alter des Gebäudes zum Betrachtungszeitpunkt angibt. Semantische Angaben geben mitunter aber auch Spielraum zur Interpretation: So könnte man aus dem Baujahr auch Rückschlüsse auf den Zustand der Gebäudesubstanz oder auf die technische Ausstattung des Gebäudes ziehen.

Daten sind einem sogenannten Lebenszyklus unterworfen, der auch mit der Aktualität der Daten zusammenhängt. Der Lebenszyklus beginnt in der Regel mit der Erhebung der Daten und schließt mit ihrer Bereitstellung ab. Jedoch sind die Objekte der Realwelt Änderungen ausgesetzt, die auch in den Daten aktualisiert werden sollten. Gegebenenfalls können die Änderungen auch zu einer Überarbeitung des Datenmodells führen.

Merkmale von Modellen

Die Beschreibung der Syntax und Semantik eines Datenmodells kann auf vielerlei Ebenen (siehe Abschnitt 2.1.3) mit unterschiedlicher Abstraktion erfolgen. Nach Stachowiak (1973) werden drei Merkmale zur Kennzeichnung von Modellen definiert [Kutzner und Eisenhut 2010, S.9].:

1. **Abbildungsmerkmal:** Ein Modell bildet ein Original der realen Welt ab. Die Beziehung zwischen Abbildung und Original ist dabei stets abhängig von der Sichtweise bzw. Auffassung des Urhebers des Modells.
2. **Reduktionsmerkmal:** Ein Modell ist eine generalisierte Abbildung des Originals. Das Modell gibt das Original nicht in Gänze wieder, sondern nur einen Teil dessen, welches der Urheber des Modells als signifikant erachtet.
3. **Pragmatisches Merkmal:** Das Modell gibt das Original in einer spezifischen Repräsentation wieder, in welcher es seinem zugedachten Zweck nach Auffassung des Urhebers nachkommt.

In Abbildung 2.2 [Staub 2009, S.8] wird ein Übergang von einem Realweltausschnitt zu einem Datenmodell grafisch verdeutlicht. Es soll ein Ausschnitt der realen Welt, mittels einer Syntax, semantisch und nach Anwendung der drei Merkmale gemäß Stachowiak (s.o.), in ein konzeptuelles Datenmodell überführt werden [Staub 2009, S.8f.]. Welche Aspekte des Realweltausschnitts zur Modellierung ausgewählt werden, hängt dabei in erster Linie von der fachlichen Sicht ab.

Begriffsbestimmung

Konzeptuelles Modell: Ein *Konzeptuelles Modell* beschreibt einen durch Abstrahierung und aus fachlicher Sicht gewählten Ausschnitt der realen oder gedachten Welt, der als Modell informell (siehe Abschnitt 2.1.4) niedergeschrieben wird. Wird das Modell formal (siehe Abschnitt

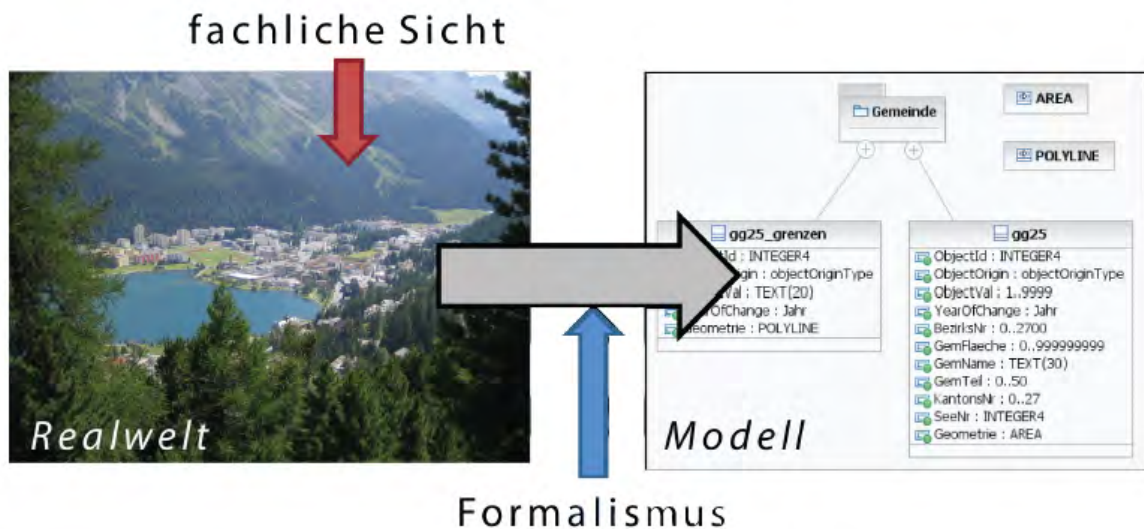


Abb. 2.2 – Modellierung eines Realweltausschnitts in einem Datenmodell [Staub 2009, S.8]

2.1.4) niedergeschrieben, so spricht man von einem *Konzeptuellen Schema*. Eine konkrete physische Implementierung wird noch nicht berücksichtigt [DIN 2015].

Auf welche Art und Weise und auf welchen Ebenen die Modellierung erfolgen kann wird im nachfolgenden Abschnitt 2.1.3 erläutert.

2.1.3 Ebenen der Datenmodellierung

Zur Datenmodellierung wird in der Geoinformatik der modellbasierte Ansatz für die Softwareentwicklung herangezogen, der darauf beruht, Aspekte der fachlichen Sicht von technischen Aspekten getrennt zu modellieren. Ein wesentliches Merkmal der Reproduzierbarkeit eines Datenmodells äußert sich in der plattformunabhängigen respektive plattformspezifischen Ausprägung des Datenmodells.

Begriffsbestimmung

Plattformunabhängig (PIM³): Die plattformunabhängige Modellierung erfolgt unabhängig und ohne Berücksichtigung von technischen Aspekten von Softwaresystemen oder Plattformen.

Begriffsbestimmung

Plattformspezifisch (PSM⁴): Die plattformabhängige oder -spezifische Modellierung erfolgt unter Berücksichtigung und mit Einbeziehung von technischen Aspekten von Softwaresystemen oder Plattformen. Sie stellt damit die Basis zur Generierung von Programmcode.

³Platform Independent Model

⁴Platform Specific Model

Möchte man eine mit geringem Aufwand verbundene Reproduzierbarkeit außerhalb einer bestimmten Softwareumgebung gewährleisten, empfiehlt es sich, ein Datenmodell zunächst plattformunabhängig zu definieren. Anschließend kann eine Spezialisierung der Modellierung plattformspezifisch mittels Transformationsregeln⁵ davon abgeleitet werden. Dadurch entsteht zwar einmalig ein Mehraufwand bei der Modellierung (da sowohl plattformspezifisch, als auch -unabhängig modelliert wird), anschließend muss das Datenmodell bei den plattformspezifischen Spezifikationen aber nicht mehr vollends neu erstellt werden.

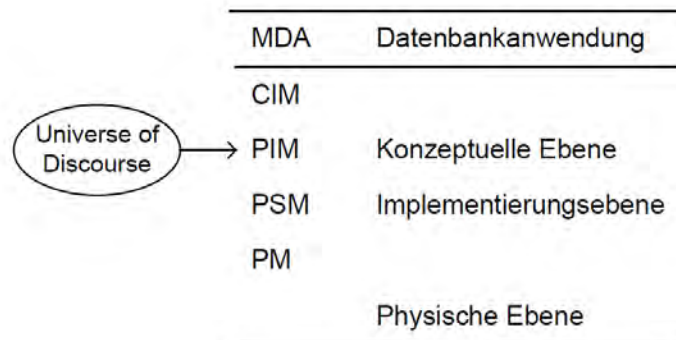


Abb. 2.3 – Ebenen der Datenmodellierung: MDA und Datenbanken [Kutzner und Eisenhut 2010, S.16]

Ein Datenmodell dient, wie bereits definiert (siehe Abschnitt 2.1.2), der syntaktischen und semantischen Beschreibung von Daten als konzeptuelles Schema. Die Realisierung eines konzeptuellen Schemas umfasst verschiedene Ebenen der Modellierung, wie sie in Abbildung 2.3 dargestellt sind [Eisenhut und Kutzner 2010, S.15f.; Unger 2012, S.10ff.]. Die Abbildung zeigt die Ebenen in der *Model-Driven Architecture* (MDA; linke Spalte) und bei Umsetzung einer Datenbankanwendung. Die *Model-Driven Architecture* wurde von der OMG „als modellbasierter Ansatz für die Softwareentwicklung“ [Kutzner und Eisenhut 2010, S.15] auf Basis der MOF⁶ entwickelt. Die Grundlage bildet das *Computation Independent Model* (CIM). Das CIM beschreibt die Anforderungen an das Modell aus fachlicher Sicht (s.o.), ohne dabei auf technische Aspekte einzugehen (in nicht-maschineninterpretierbarer Form). Somit stellt es das Grundkonzept dar, mit größtem Grad an Abstraktion in Bezug auf das Datenmodell bzw. geringstem Grad an Abstraktion in Bezug auf die Realwelt. Eine Ebene darunter ist das *Platform Independent Model* (PIM) anzusiedeln. In dieser Modellierungsebene wird das Modell bereits konkreter, allerdings nach wie vor unabhängig von Plattformen beschrieben (z.B. durch UML⁷-Klassendiagramme). Dem gegenüber steht aus Sicht von Datenbankanwendungen die *konzeptuelle Ebene*, welche ebenfalls plattformunabhängig die Struktur dessen was erfasst werden soll beschreibt. Das *Platform Specific Model* (PSM) spezialisiert das PIM zu

⁵Transformationsregeln sind Regeln, welche die Schritte bei der Überführung eines konzeptuellen Schemas in ein Transformat beschreiben [Kutzner und Eisenhut 2010, S.25].

⁶Die **Meta Object Facility** (MOF) ist die Spezifikation einer Meta-(Meta-)Daten-Architektur zur Beschreibung von PIMs (s.o.) und somit die unterste und sich selbst definierende Ebene der Datenmodellierung [Kutzner und Eisenhut 2010, S.31f.].

⁷Die **Unified Modeling Language** ist eine Modellierungssprache zur konzeptuellen Gestaltung von Datenmodellen.

einem plattformspezifischen Modell, indem es die Erweiterungen und Besonderheiten eines Systems bzw. einer Plattform miteinbezieht (z.B. ein XML⁸-Schema). Ebenso handelt es sich bei der *Implementierungsebene* von Datenbankanwendungen um eine plattformspezifische Umsetzung. Wird das Konzept schließlich in einen Programmcode umgesetzt (z.B. ein GML⁹-Instanzdokument), dann spricht man in der letzten Ebene vom *Platform Model* (PM). Darunter noch wird bei Datenbankanwendungen die *physische Ebene* beschrieben, um Strukturen wie Verweise (Zeiger) auf Variablen oder Indexierung für ein gesteigertes Leistungsvermögen zu optimieren.

Wenn bislang von Modellen die Rede war, so zumeist von konzeptuellen Datenmodellen. Tatsächlich existieren verschiedene Arten von Modellen die unterschiedliche Konzepte verfolgen [Kutzner und Eisenhut 2010, S.21ff.]. Zwei dieser Modelle seien an dieser Stelle kurz erläutert:

- **Anwendungsschema**

Das Anwendungsschema ist ein konzeptuelles Schema aus fachlicher Sicht zur semantischen Beschreibung von Geodaten. Es wird nach der ISO¹⁰ 19101 als *conceptual schema* definiert [DIN 2015]. Ein Anwendungsschema ermöglicht die Spezifikation eines eigenen bzw. fachbezogenen Vokabulars und eigener Deklarationen von Objekten. Das Anwendungsschema AAA 6.0 CityGML 1.0 der AdV definiert beispielsweise eigene Kodierungslisten (siehe Abschnitt 3.1.3), wodurch es sich vom ursprünglichen CityGML 1.0 abgrenzt. Die anwendungsspezifischen Eigenschaften werden im Namensraum des Anwendungsschemas definiert, der sich vom ursprünglichen Namensraum aufgrund der Spezifizierung unterscheidet. Hingegen ist ein GML-Anwendungsschema ein XML-Schema, mit dem ein Anwendungsfall auf Ebene der Transferformate semantisch und syntaktisch beschrieben wird.

- **Implementierungsschema**

Das Implementierungsschema beschreibt die plattformspezifische Modellierung (PSM), also die Umsetzung eines PIM in einem konkreten Plattform, aus dem ein Transferformat erzeugt werden kann [Kutzner und Eisenhut 2010, S.24], z.B. das Transferformat GML (s.u.).

Die OMG definiert eine *4-Schichten-Architektur*: Mit den Ebenen M0 (Info Layer) zur Wiedergabe der Daten; M1 (Model Layer) zur Beschreibung der Daten mit einem Modell; M2 (Metamodel Layer) zur Beschreibung von Semantik und Syntax mit einer Modellierungssprache; und M3 (Metametamodel Layer) zur Definition der Modellierungssprache [Kutzner und Eisenhut 2010, S.22f.].

Datenmodelle und Anwendungsschemata werden also auf Modellebene (M1) erzeugt. Aus diesen können anschließend Formate zur plattformspezifischen Beschreibung abgeleitet werden. Diese Ebe-

⁸Die **Extensible Markup Language** (XML) ist eine Auszeichnungssprache zur Darstellung von Daten (siehe Abschnitt 2.1.4).

⁹Die **Geography Markup Language** (GML) ist eine Auszeichnungssprache zur Darstellung von Geodaten (siehe Abschnitt 2.1.4).

¹⁰**I**nternational **O**rganization for **S**tandardization (ISO)

ne wird auch Formatebene genannt, obwohl sie noch der Modellebene angehört. Aus den Formaten können schlussendlich Daten (sogenannte Instanzen) erzeugt werden. Man spricht an dieser Stelle von der Daten- oder Instanzebene (M0). Die tatsächlichen physischen Daten entsprechen somit den Instanzen eines Datenmodells, modelliert unter Anwendung einer Modellierungssprache.

2.1.4 Modellierungssprachen

Modellierungssprachen sind Sprachen, mit denen Modelle gebildet und definiert werden können. In Kutzner und Eisenhut (2010, S.10ff.) erfolgte eine Beschreibung von Sprachen anhand von vier Eigenschaften, welche die Charakterisierung ihrer Ausdrucksform darstellen: Formal, informell, visuell und textuell. Formal bedeutet, dass ein Modell oder Programmtext gebildet werden soll, während dies bei einer informellen Ausdrucksform nicht der Fall ist (z.B. bei einer gesprochenen Sprache). Demzufolge sind Modellierungssprachen stets formaler Natur. Die Begriffe visuell und textuell sind selbsterklärend. In den vorherigen Abschnitten wurden bereits diverse Ausprägungen von Modellen erwähnt: UML-Modell, XML-Schema und (XML- und) GML-Instanzdokumente. Datenmodelle werden konzeptuell und plattformunabhängig (PIM) beschrieben, um sie einer Vielzahl an Nutzern zur Weiterverwendung oder Spezialisierung durch Erweiterung/Eingrenzung zur Verfügung zu stellen (s.o.). Eine häufig genutzte und standardisierte Modellierungssprache ist UML.

- **Unified Modeling Language (UML)**

Die Modellierung mit UML erfolgt objektorientiert (siehe Abschnitt 2.5) und visuell. Das bedeutet: Es werden grafisch Diagramme nach festgelegten Regeln erstellt, die Objekte mit deren Attributen und Hierarchien, Strukturen und Aktivitäten zwischen den Objekten darstellen. UML ist sehr umfangreich, da es sich über alle Fachbereiche der Informatik erstreckt und zudem kann mittels Profilen eine spezialisierende Einschränkung von Modellen erfolgen (siehe Abschnitt 2.3).

Um die im Datenmodell beschriebenen Datensätze zu transferieren, wird auf geeignete Transferformate (z.B. GML) zurückgegriffen. Darin wird die in den UML-Modellen beschriebene Semantik aufgegriffen und auf die konkret physisch vorhandenen Objekte (Instanzen) abgebildet. Hierfür wird aus den UML-Modellen zunächst ein XML-Schema erzeugt, welches die Semantik und Syntax in die Formate XML bzw. GML überträgt und die Instanzierung ermöglicht. Abbildung 2.4 zeigt diesen Sachverhalt. Es ist jedoch nicht erforderlich, ein UML-Modellen zu erzeugen und daraus die XML-Schema-Dokumente (XSD) abzuleiten. Diese können auch direkt selbst erzeugt werden, ohne explizite UML-Modellierung. Allerdings kann die Ableitung aus UML-Profilen unter Anwendung von Kodierungsregeln automatisch erfolgen.

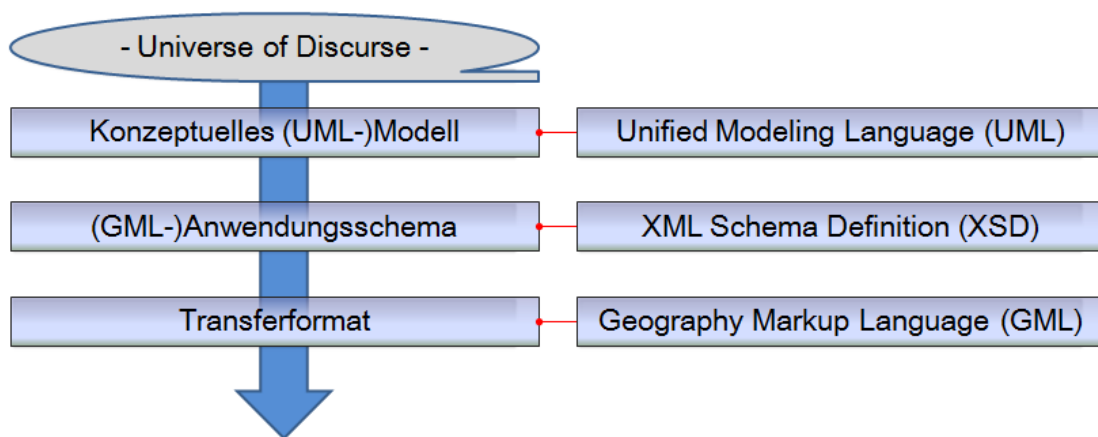


Abb. 2.4 – Hierarchie der Modellierungssprachen bei der Erzeugung von GML-Instanzen

- **Extensible Markup Language (XML)**

Die Extensible Markup Language ist eine Auszeichnungssprache (engl. Markup Language) zur formalen und textuellen Speicherung und Wiedergabe von Daten. Sie wurde vom World Wide Web Consortium (W3C)¹¹ entwickelt. Es handelt sich um eine Metasprache, die durch eigene Definitionen von Namensräumen und Elementnamen zu einem Anwendungsschema spezialisiert werden kann. Durch ihre geschachtelte Textstruktur wird eine Hierarchie unter den Objekten wiedergegeben (siehe Abschnitt 2.5) [Staub 2009, S.16f.]. Das zugrunde liegende Sprachparadigma wird als XML-Paradigma bezeichnet [Kutzner und Eisenhut 2010, S.17].

- **Geography Markup Language (GML)**

Bei der Geography Markup Language handelt es sich um einen XML-Dialekt (erweitert um Namensräume und Elementnamen), speziell entwickelt zur Speicherung und Wiedergabe von Geodaten. Die Erweiterungen zu den XML-Basistypen umfassen in erster Linie Objekte und Begriffe aus der Geoinformatik. Auch mit GML ist es möglich, anwendungsspezifische Profile zu entwickeln (z.B. OGC GML Simple Features Profile) um eine Spezialisierung vorzunehmen [Staub 2009, S.17]. Das Schema von GML-Dokumenten wird mit XML-Schemata beschrieben und basiert auf dem objektorientierten Paradigma (siehe Abschnitt 2.5).

2.2 Relevante Normen, Spezifikationen und Vorschriften

Normen, Spezifikationen und Vorschriften in der Informatik haben für gewöhnlich zum Ziel, Standards und Routinen für u.a. Daten, Modelle und Dienste zu entwickeln. Sowohl Datenanbietern - bei denen es sich oftmals um staatliche Behörden handelt - als auch Softwareherstellern ist daran gelegen, sich technisch an diese Normen etc. zu halten und eigene (plattformspezifische) technische Aspekte

¹¹Das **World Wide Web Consortium (W3C)** ist ein Gremium zur interoperablen Standardisierung von technischen Spezifikationen und Richtlinien [Wikipedia 2015b].

unter deren Berücksichtigung zu implementieren: Denn durch interoperable (s.u.) Zusammenarbeit können beiderseits Synergieeffekte¹² genutzt werden.

In der Geoinformatik hat sich als Standard die Reihe ISO 19100 etabliert. Beispielsweise werden mit der Norm *ISO 19109 Geographic Information - Rules for application schema* die Regeln zur Erstellung von Anwendungsschemata (siehe Abschnitt 3.1) definiert [Kutzner und Eisenhut 2010, S.33]. Im Grunde wird mit der Reihe das gesamte Spektrum von der Definition von Begrifflichkeiten (z.B. informell als konzeptuelles Modell niedergeschriebener Realweltausschnitte in *ISO 19101 - Geographic information - Reference model* [Kutzner und Eisenhut 2010, S.10]) bis zum Datenaustausch erfasst. Eine Übersicht zeigt die - aus Staub (2009, S.45) entnommene - Abbildung 2.5 der wichtigsten Normen und Standards.

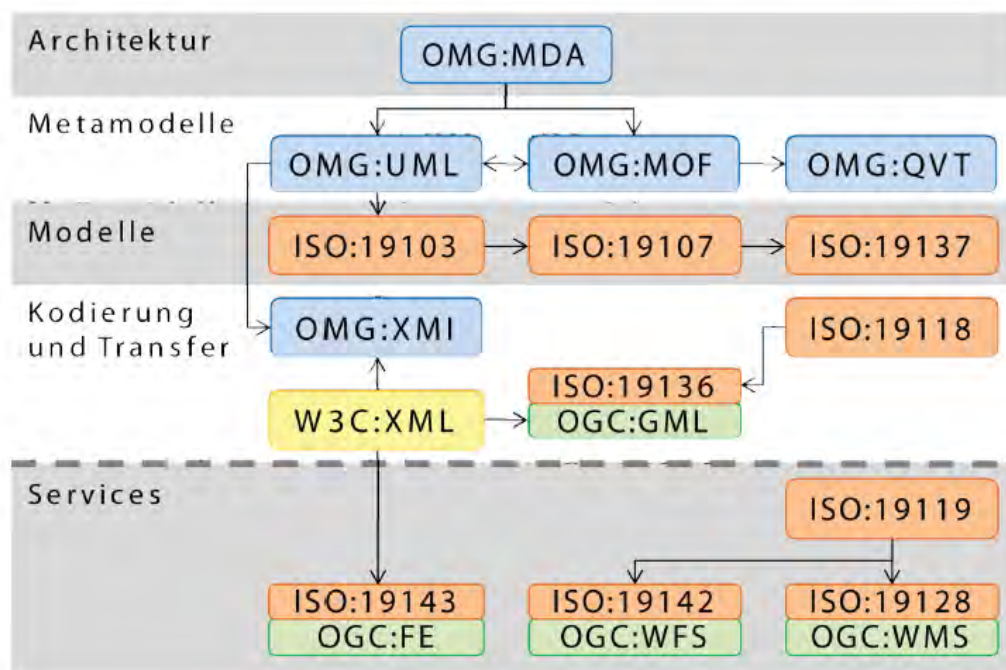


Abb. 2.5 – Systematische Übersicht über einschlägige Normen und Standards [Staub 2009, S.45]

In der Abbildung erfolgt die Anordnung der Normen und Standards „nach Abstraktionsniveau beziehungsweise nach Funktionalität“ [Staub 2009, S.45]. Jedes Kästchen repräsentiert eine Norm bzw. einen Standard, wobei das Kürzel vor dem Doppelpunkt den Urheber angibt (z.B. OGC).

In Fichtinger (2011, S.13ff.) erfolgt eine ausführliche Auflistung „relevanter Normen, Standards und Spezifikationen“ im Kontext der Datenmodelltransformation.

¹²Synergien beschreiben das Zusammenwirken von Faktoren (und Systemen) in einer sich gegenseitig förderlichen Art und Weise [Wikipedia 2015c].

2.2.1 Interoperabilität

Einen wichtigen Grundsatz in der Modellierung von Datenmodellen stellt die Interoperabilität dar.

Begriffsbestimmung

Interoperabilität: Interoperabilität ist im Falle von Geodatenätzen die Fähigkeit ihrer möglichen Kombination und im Falle von Diensten ihrer möglichen Interaktion, ohne wiederholtes manuelles Eingreifen und in einer Weise, dass das Ergebnis (Produkt) kohärent ist und hinzugefügte Werte die Datensätze und Dienste um einen Mehrwert erweitern [Europäisches Parlament und Rat 2007, Art.3; INSPIRE JRC 2014, S.11; Staub 2009 S.22; Fichtinger 2011, S.11f.].

In Staub [2009, S.24ff.] wird zwischen syntaktischer und semantischer Interoperabilität unterschieden. Ersteres „(...) baut auf die standardisierten (Internet-)Kommunikationsprotokolle und Datentransformate (...)“ [Staub 2009, S.24, verändert] im Anbietermodell¹³ auf, wie sie von Gremien, Vereinen und Konsortien (s.o.) entwickelt wurden/werden. Die semantische Interoperabilität „(...) zeichnet sich (...) dadurch aus, dass zunächst auf konzeptioneller, systemunabhängiger Ebene die Datenmodelle durch semantische Modelltransformationen (...) aufeinander abgebildet werden und die Daten danach (automatisch) durch eine entsprechende Transformationssoftware transformiert werden. Die Datenstrukturen der verfügbaren Daten sind dem Nutzer also zugänglich beziehungsweise manipulierbar (...)“ [Staub 2009, S.25].

2.2.2 INSPIRE

Am 15. März 2007 wurde vom Europäischen Parlament und Rat die Richtlinie „zur Schaffung einer Geodateninfrastruktur in der Europäischen Gemeinschaft (INSPIRE¹⁴) in Kraft gesetzt. Ausgangspunkte zur Bildung der Initiative waren in erster Linie Heterogenitäten in der „großen Vielfalt von Formaten und Strukturen für die Verwaltung von Geodaten in der Gemeinschaft“ [Europäisches Parlament und Rat 2007, S.2 (16), verändert] und Schwierigkeiten „bei der Suche nach bestehenden Geodaten und der Prüfung ihrer Eignung für einen bestimmten Zweck“. Ergo sollen die hieraus „entstehenden Zeit- und Ressourcenverluste“ [Europäisches Parlament und Rat 2007, S.2 (15), verändert] in der Verwaltung und Planung zur Kostensenkung reduziert werden. Stattdessen sollten nun die „Geodateninfrastrukturen der Mitgliedstaaten (...) so ausgelegt sein, dass Geodaten (...) auf kohärente Art verknüpft und von verschiedenen Nutzern und für verschiedene Anwendungen genutzt werden können (...)“ [Europäisches Parlament und Rat 2007, S.2 (6)]. Also im Sinne einer gemeinschaftlichen und einheitlichen Geodateninfrastruktur auf europäischer Ebene (EGDI¹⁵). Somit ist eines der wesentlichen Ziele von INSPIRE die Förderung der Interoperabilität innerhalb Europas.

Die INSPIRE-Richtlinie wurde in Deutschland unter Berücksichtigung der Gesetzgebungskompetenzen zwischen Bund und Ländern mit einem Bundesgesetz (siehe Abschnitt 2.2.3 [GeoZG 2009])

¹³Das Anbietermodell richtet sich nach den Anforderungen des Anbieters, nicht nach denen des Nutzers; ein Nutzer muss seine Strukturen also gegebenenfalls anpassen.

¹⁴Infrastructure for **S**Patial **I**nfo**R**mation in the **E**uropean Community (INSPIRE)

¹⁵European **G**eospatial **D**ata Infrastructure (EGDI)

und 16 Ländergesetzen (siehe Abschnitt 2.2.4 [u.a. BayGDIG 2008]) in nationales Recht umgesetzt. Durch das Inkrafttreten der Gesetze wurden die geodatenhaltenden Behörden dazu verpflichtet, eine Geodateninfrastruktur vorzuhalten, die eine Geodatenbasis entsprechend den definierten Konformitätsanforderungen der Durchführungsbestimmungen zur INSPIRE-Richtlinie bereitstellt. In einem ersten Schritt zur Umsetzung der Verordnungen mussten die Geodaten in ihren bereits bestehenden Datenmodellen über INSPIRE-konforme Darstellungs- und Downloaddienste online zugänglich gemacht werden. In den weiteren Schritten wird gefordert, die Bereitstellung der Download- bzw. Datendarstellungsdienste endgültig und vollumfänglich konform zur INSPIRE-Richtlinie anzubieten [Europäisches Parlament und Rat 2007]. Die Fristen, innerhalb derer die Bereitstellung erfolgen muss, unterscheidet für die Geodaten-Themen der Anhänge. Bis 23.11.2017 müssen die Daten zum Anhang I bereitgestellt werden (u.a. *Protected Sites*) und bis zum 21.10.2020 die Daten zum Anhang III (u.a. *Buildings*). Es ist nicht vorgesehen eine Neuerhebung von Geodaten durch die Initiative zu veranlassen [Europäisches Parlament und Rat 2007, S.2 (13)].

Aufbau von INSPIRE

Als Grundlage zur Realisierung der Richtlinie aus technischer Sicht wurden rechtlich wirksam fünf Durchführungsbestimmungen (engl. Implementing Rules) zur Richtlinie als formale EG-Verordnungen erlassen. Die Durchführungsbestimmungen gliedern sich in fünf relevante Themenbereiche, die zur Umsetzung der Richtlinie von Bedeutung sind [Fichtinger 2011, S. 22; Geoportal GDI-DE 2015a]:

- **Metadaten**

Diese Durchführungsbestimmung definiert die abstrahierte Beschreibung der Geodatensätze und -dienste auf Metaebene, um Informationen zu den Eigenschaften und den Verwendungszweck der Datensätze zu liefern und deren Auffindung innerhalb der Geodateninfrastruktur zu ermöglichen [Geoportal GDI-DE 2015b].

- **Netzdienste**

Hier werden Bedingungen über online für den Nutzer zugängliche Geodatendienste bezüglich Suche der Geodatensätze und deren Leistungsvermögen beim Download, bei Transformationen und bei der Anzeige festgelegt [Geoportal GDI-DE 2015c].

- **Monitoring und Berichtswesen**

In dieser Durchführungsbestimmung werden Indikatoren zur Überwachung des Fortschritts der Umsetzung von INSPIRE durch die Mitgliedsstaaten und für das Qualitätsmanagement während des Betriebs der Geodatensätze und -dienste niedergeschrieben. Die Berichterstattung durch die Mitgliedsstaaten muss demnach jährlich erfolgen [Geoportal GDI-DE 2015d].

- **Gemeinsame Nutzung von Daten**

Die Durchführungsbestimmung enthält Regelungen zu Lizenzierung der Zugriffsrechte, Nutzungsbedingungen und dem Urheberrecht [Geoportal GDI-DE 2015e].

- **Interoperabilität von Geodatenätzen und -diensten**

Die Durchführungsbestimmung legt die *Data Specifications* (dt. (Daten-)Spezifikationen) zur Definition und Umsetzung der Geodaten-Themen in den Anhängen I bis III der Richtlinie fest [Geoportal GDI-DE 2015f].

Zur konkreten technischen Umsetzung oblag die Entwicklung eines technischen Grundlagenrahmenwerks zu INSPIRE den sogenannten *Drafting Teams*¹⁶, die zur Ausarbeitung der INSPIRE-Dokumente gebildet wurden. Darin enthalten sind die *Technical Guidance*-Dokumente, welche die technischen Ausprägungen zu den Durchführungsbestimmungen formulieren (z.B. zur Implementation der Downloaddienste) und die Regelwerke der Datenmodellierung festlegen. In der Praxis stellt die interoperable Datenmodellierung den Kern der Direktive dar. Die Interoperabilität der Geodatenätze wird vor allem dadurch realisiert, dass die Vorgaben zu den zur Verfügung zu stellenden Geodatenätzen ebenfalls durch die Direktive vorgegeben werden.

Der Aufgabenbereich zur Gestaltung der *Data Specifications* innerhalb des technischen Rahmen-

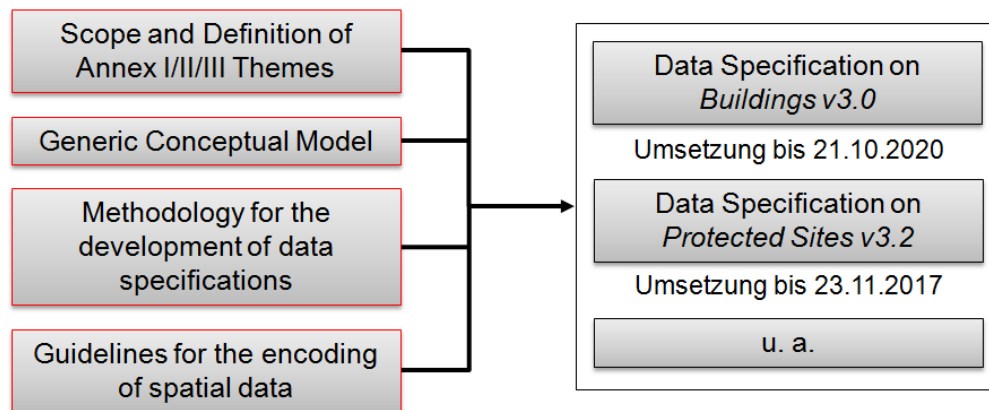


Abb. 2.6 – Rahmenwerk zur Entwicklung der INSPIRE *Data Specifications*

werks wird durch vier Dokumente vorgegeben (Abbildung 2.6). Diese bestimmen die technischen Grundlagen zur Gewährleistung der einheitlichen Entwicklung der Spezifikationen [Fichtinger 2011, S. 26]: INSPIRE *Scope and Definition of Annex I/II/III*, INSPIRE *Generic Conceptual Model*, INSPIRE *Methodology for the development of data specifications* und INSPIRE *Guidelines for the encoding of spatial data*.

2.2.3 Gesetzgebung auf Ebene der Mitgliedsstaaten

Um zu verdeutlichen, wie sich die auf europäischer Ebene festgelegten rechtlichen Grundlagen auf die Mitgliedsstaaten auswirken, soll noch am Beispiel Deutschland kurz auf die Rechtssetzung der

¹⁶Drafting Teams (engl.) sind Redaktions- und Arbeitsgruppen.



Abb. 2.7 – Die Geodaten-Themen der Anhänge I bis III von INSPIRE

Mitgliedsstaaten eingegangen werden. Auf Bundesebene wurde 2009 das *Gesetz über den Zugang zu digitalen Geodaten* (Geodatenzugangsgesetz - GeoZG) verabschiedet [GeoZG 2009]. Ziel des Gesetzes nach GeoZG (2009) Paragraph 1, ist der „Aufbau einer nationalen Geodateninfrastruktur“, um „den rechtlichen Rahmen für (1) den Zugang zu Geodaten, Geodatendiensten und Metadaten von geodatenhaltenden Stellen sowie (2) die Nutzung dieser Daten und Dienste (...)“ zu setzen.

Das Bayerische Geodateninfrastrukturgesetz (BayGDIG) vom 22. Juli 2008 ist das Resultat von INSPIRE auf Landesebene in Bayern [BayGDIG 2008]. Als Ziel wird hier auch die Schaffung eines rechtlichen Rahmens „für den Ausbau und den Betrieb einer Geodateninfrastruktur“ auf Landesebene ausgegeben [BayGDIG 2008, Art.1]. Die Geodateninfrastruktur auf Landesebene bildet eine Teilmenge der Geodateninfrastruktur Deutschlands und wird in Kurzform als GDI-BY bezeichnet.

2.3 Datenmodellierung für Geodaten

Geodaten repräsentieren im Generellen digitalisierte Objekte und/oder Zustände unserer physisch realen Welt, indem sie diese logisch respektive mathematisch verwertbar darstellen, visualisieren und georeferenzieren¹⁷. Damit werden modellierte Objekte bzw. Sachverhalte - theoretisch zu jedem Zeitpunkt - räumlich darstellbar.

¹⁷Georeferenzierung beschreibt die Verknüpfung von digitalen Objekten zu ihren realweltlichen Objekten, z.B. über die Zuweisung von Ortskoordinaten zu den Daten.

Bei Geo-Objekten (d.h. Objekten der Realwelt) handelt es sich oftmals um Gebilde, die sich zu keiner strengen, oder einer sehr detailreichen geometrischen Anordnung zusammensetzen und mitunter in komplexe Wechselwirkungen mit anderen Gebilden verstrickt sind. Wechselwirkungen werden auch dadurch auf abstrakte Weise hervorgerufen, dass bei der Nutzung häufig auf die Kombination von Geodaten zurück gegriffen wird (z.B. die Kombination von Fachdaten wie Versorgungsnetze mit einer digitalen Kartengrundlage) [Donaubauer 2004, S. 13]. Eine digitale Abbildung der Objekte erfordert nach zuvor genannten Prinzipien (siehe Abschnitt 2.1.2) jedoch eine Reihe von Einschränkungen, die einige Probleme bei der Modellierung aufwerfen (u.a.):

- Die geometrische Beschreibung natürlicher Objekte (z.B. Flussläufe) ist komplex und erfordert je nach Detailgrad der Modellierung einen hohen Aufwand bei der Datenerfassung und -aufarbeitung.
- Die geometrische Beschreibung der Form ist nicht eindeutig und kann oft auf vielfältige Art und Weise erfolgen. Es muss ein Mittelmaß zwischen Aufwand und korrekter Repräsentation gefunden werden.
- Bei zu geringem Detailgrad der geometrischen Darstellung bzw. zu grober Rasterung der verwendeten Referenzpunkte des Objektes gehen zu kleine Details verloren (Aliasing-Effekt).
- Die Abgrenzung von Einzelobjekten (z.B. Vegetation im Verbund) kann schwierig sein und zusätzlich die Automatisierung erschweren.
- Der Umfang der Objektrepräsentation muss den Nutzungsanforderungen genüge tun und zugleich der Interoperabilität dienen. Wird wie im Fall von INSPIRE eine weitläufige Vereinheitlichung angestrebt, bei gleichbleibender Vielfalt in den Anforderungen und Anwendungen seitens der Nutzer, müssen die Datenmodelle auch entsprechend umfangreich sein. Andererseits aber auch variabel, um die Datensätze an die vorgesehene individuelle Nutzung anpassen zu können und somit das Datenvolumen möglichst klein halten zu können.

Da die Definition und Realisierung eines Datenmodells stark von den Nutzungsanforderungen abhängt, variieren entsprechend die angebotenen Datenmodelle von Anbieter zu Anbieter.

Im Vorfeld der Erstellung der Datenspezifikation für INSPIRE wurden zahlreiche Nutzeranforderungen an die Spezifikationen gesammelt und zur Auswertung katalogisiert. In Abbildung 2.8 wurde das Ergebnis für die *Data Specification on Buildings* dargestellt: Die selektierte und harmonisierte Auswahl der Nutzeranforderungen, die schließlich in den Feature Types (in der Grafik: fettgedruckt - satte Farben) und Attributen (in der Grafik: normalgedruckt - bleiche Farben) umgesetzt wurden. Wobei sich die Anforderungen in vier Sektionen gliedern, unterschieden nach Harmonisierung auf (1) europäischer oder (2) lokaler Ebene und (3) häufiger Verfügbarkeit bzw. (4) rarer Verfügbarkeit von Daten. Nach dieser Unterteilung wurde auch die Datenspezifikation mit modularen Profilen gestaltet, sodass je nach Anforderungen das geeignetste Profil verwendet werden kann [INSPIRE DS BU 2013, S. 3ff.].

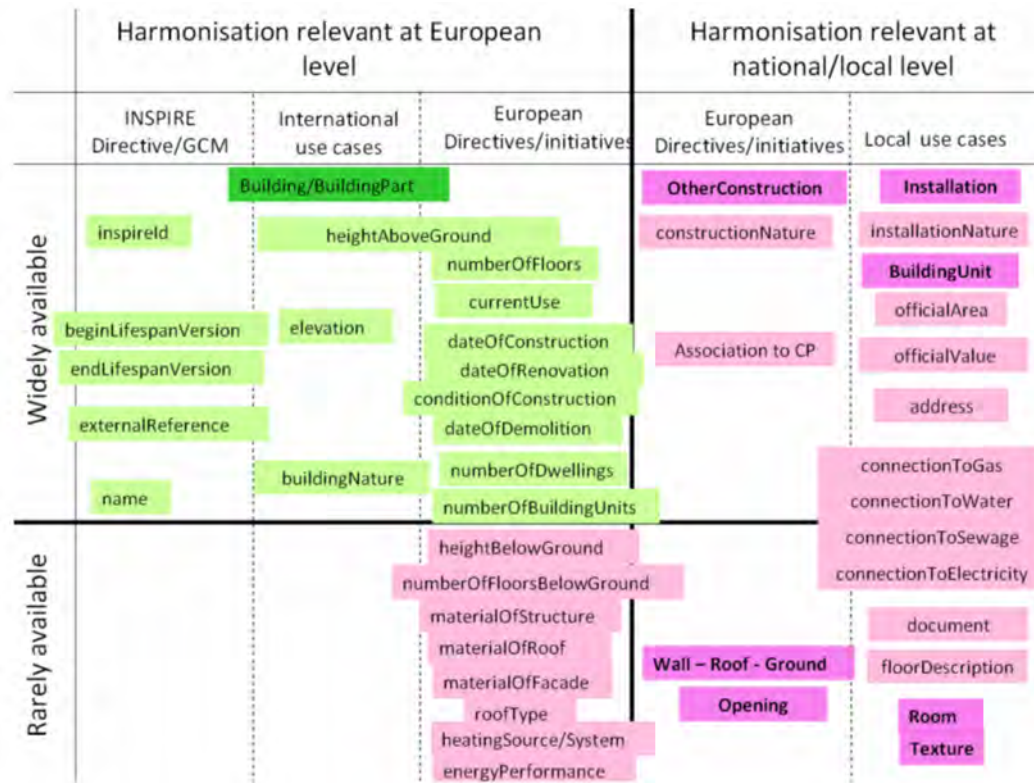


Abb. 2.8 – Semantische Nutzeranforderungen an CityGML für INSPIRE [INSPIRE DS BU 2013, S. 4]

2.3.1 Heterogenität von Geodaten

In Gedrange und Neubert (2011, S. 2ff.) wird die Heterogenität von Geodaten analysiert. Als Ursache können vielerlei Gründe ausgemacht werden. Beispielsweise führen durch Tradition begründete unterschiedliche Sichtweisen in der Folge zu unterschiedlichen Datenmodellen. Es kann dabei zwischen semantischer und geometrischer Heterogenität unterschieden werden.

Unter semantischer Heterogenität wird die unterschiedliche Bedeutungszuweisung zu Geoobjekten aufgrund von „fehlender Übereinkunft bezüglich der Bedeutung von Dingen“ verstanden, wie z.B. „terminologischen Abweichungen zwischen Disziplinen oder sozialen Gruppen“ [Gedrange und Neubert 2011, S.3].

Das Auftreten von geometrischer Heterogenität kann bei der Beschreibung von Geodaten beobachtet werden, welche den gleichen Realweltauusschnitt beschreiben, jedoch von unterschiedlichen Urhebern gestalten wurden. Diese Form der Heterogenität kann in unterschiedlichen zur Anwendung kommenden Koordinatenreferenzsystemen begründet sein oder in Differenzen bei der Darstellung der Objekte (z.B. unterschiedliche Detaillierungsgrade oder Generalisierungsmethoden). Unterschiedliche Methoden der Datenerhebung oder Modellierung schlagen sich ebenfalls in Abweichungen zwischen Datensätzen nieder. Im Kontext von INSPIRE sind geometrische Heterogenitäten insbesondere dann von Belang, wenn Geo-Objekte von verschiedenen Datenanbietern redundant bereitgestellt werden und beide Datensätze miteinander in eine Prozessierung miteinbezogen werden sollen. Dies ist z.B. in Gebieten nahe von Staatsgrenzen der Fall.

Nach Fichtinger (2011, S.45) ist „die Analyse und Beschreibung der semantischen Heterogenität zwischen Datenmodellen (...) eine der Grundlagen für semantische Transformationen“.

2.3.2 Geodateninfrastrukturen

Geodaten sind vor allem dann vollumfänglich und optimal nutzbar, wenn sie zugänglich und organisiert angeboten werden, auf qualitative Hochwertigkeit geprüft wurden und eine kombinierte Nutzung möglich ist. Alle diese Aspekte zusammengekommen lassen sich durch die Bildung einer Geodateninfrastruktur (GDI) verwirklichen. „Eine Geodateninfrastruktur schafft technische, organisatorische und administrative Grundlagen für die gemeinsame Nutzung, die Zugänglichkeit und die Verwendung von interoperablen Geodaten und Geodatendiensten“ [GDI-BY 2015; Fichtinger 2011, S.11]. Wie in Abbildung 2.9 schematisch dargestellt, fundiert die technische Struktur einer GDI auf verbindlichen Rechtsnormen (z.B. GeoZG), Normen und Standards (z.B. ISO Normen), sowie Vereinbarungen über den Zugang und die Nutzung der Daten bzw. Dienste. Die Daten selbst werden in Datenbanken gespeichert und via Datendienste bereitgestellt, wozu Netzdienste implementiert sein müssen (z.B. Schnittstellen zur Kommunikation und Datenaustausch über das Internet). In Metadaten werden Informationen über die Datensätze und -dienste veröffentlicht. Mittels Koordinierungs- und Überwachungsmechanismen können Qualität und Aktualität der Daten und Dienste gewährleistet werden. Grundsätzlich steht das Qualitätsmanagement in Wechselwirkung mit den getroffenen Vereinbarungen, aufgrund von technischen Weiterentwicklungen in Normen und Standards, die ihrerseits wiederum durch praktische Erfahrungen initiiert werden.

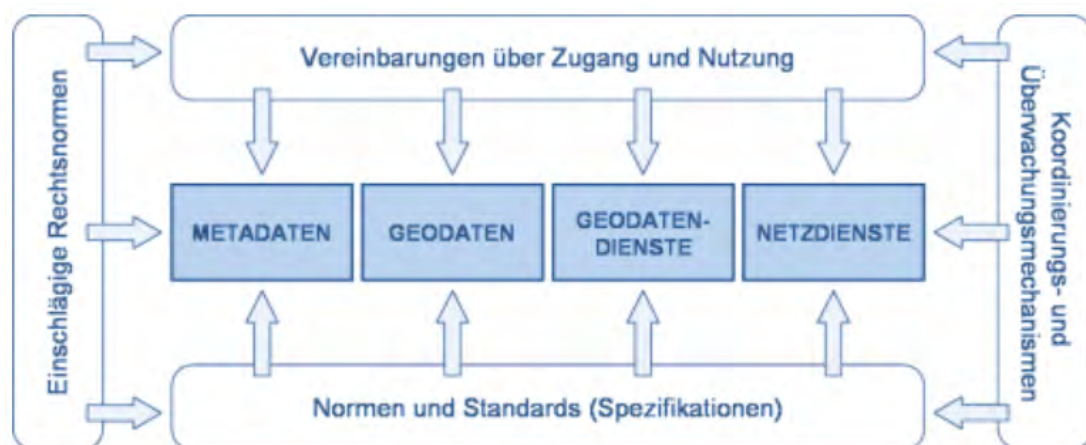


Abb. 2.9 – Aufbau der GDI-BY [GDI-BY 2015]

2.3.3 Datentransfer

Beim Datentransfer handelt es sich um die Übertragung von Daten zwischen Systemen unter Erhaltung von Syntax und Semantik. Die Übertragung kann nur auf Ebene der Datenformate stattfinden,

da diese als einzige Ebene der Modellierung (Ebene M0 in der 4-Schichten-Architektur) die konkreten Objektinstanzen abbildet. Als Transferformat wird in der Geoinformatik häufig GML (siehe Abschnitt 2.1.4) eingesetzt. Es ist zu beachten, dass auch bei den Kodierungsschritten während der Überführung von konzeptuellem Modell (M1) über die Formatebene (M1) in die Transferformate die Semantik korrekt überführt wird [Kutzner und Eisenhut 2010, S.24f.].

In GDIs fungieren Transferformate üblicherweise als Ausgabeformate der Netzdienste, beispielsweise als Ausgabeformat für Daten einer Abfrage durch einen Web Feature Service¹⁸ (WFS). In der ISO-Reihe wird der Datentransfer in der Norm ISO 19118 abgehandelt.

2.4 Semantische Datenmodelltransformation und Abbildungssprachen

Bei einer Transformation handelt es sich um eine gezielte/gesteuerte Änderung oder Umwandlung des Zustands von (digitalen) Objekten und Strukturen. Dadurch ist eine Transformation auch von einem Transfer abzugrenzen, bei dem diesbezüglich keine Änderungen erfolgen.

Laut Kutzner und Eisenhut (2010, S.25f.) können „mittels *Modelltransformation* (...) aus vorhandenen Modellen neue Modelle erzeugt werden und es können aus vorhandenen Modellen auch Code, Daten oder Transferformate generiert werden“, wobei die „Modelltransformation an sich (...) unabhängig von einem bestimmten Paradigma“ ist. Bei der im vorherigen Abschnitt (2.3.3) genannten Überführung zwischen den Modellierungsebenen M1 bis M0 handelt es sich also um Modelltransformationen. Da diese Modelle Geodaten abbilden, spricht man synonym von einer Datenmodelltransformation.

2.4.1 Semantische Datenmodelltransformation

Die Datenmodelltransformation wird zwischen einem Quell- und Zielmodell durchgeführt, indem mittels einer Transformationssprache Transformationsregeln definiert und eingesetzt werden [Kutzner und Eisenhut 2010, S.26]. Die Quell- und Zielmetamodelle müssen dabei nicht konform zueinander sein und die Transformation wird auf den konzeptuellen Modellen [Zedlitz 2013, S.12; Kutzner und Eisenhut 2010, S.26] oder auf einer darunter angeordneten Hierarchiestufe angewandt, beispielsweise auf Formatebene. Mit den Transformationsregeln wird das Quellmodell in das Zielmodell überführt. Dies ist so zu verstehen, dass kein gänzlich neues Modell generiert wird, sondern die Semantik des Quellmodells im Zielmodell differenziert ausgedrückt und umstrukturiert, aber konform wiedergegeben wird. Die Semantik muss bei einer semantischen Datenmodelltransformation erhalten bleiben. Eine syntaktische Transformation beschreibt die Umstrukturierung der Syntax eines Modells, wie bei der zuvor erwähnten Erzeugung der Datenebene (M0) aus der Modellebene (M1).

¹⁸Ein **Web Feature Service** (WFS) ist ein internetbasierter Dienst, mit dem für Nutzer Geodaten (Vektordaten) bereit gestellt werden können [GDI-BY WFS 2013, S.5].

Prinzipiell kann eine Abbildung auch direkt zwischen den Daten vorgenommen werden, ohne ein Schema oder konzeptuelles Modell miteinzubeziehen, allerdings nur verbunden mit hohem Aufwand und einem hohen Maß an manuell durchgeführten Arbeitsschritten. Außerdem ist diese Form der Abbildung auch sehr anfällig für Fehler. Bequemer und sicherer lassen sich die Transformationen auf Ebene der Transferformate mit einem Datenmodelltransformationstool (ETL-Tool) realisieren, wie in Abbildung 2.10 visualisiert. Jedes ETL-Tool bedient sich einer anderen Abbildungssprache (siehe Abschnitt 2.4.2), um die Transformationsregeln (syn. Abbildungsregeln) zu definieren. Zudem besteht meist die Möglichkeit, die Quelldaten einzulesen und auf diese die Abbildungsregeln anzuwenden um sie in die Zieldaten zu überführen. Das macht letztlich einen elementaren Arbeitsschritt bei der Bereitstellung der Daten durch eine GDI aus.

Die Transformation zwischen Datenmodellen wird als *semantische Transformation* bezeichnet [Kutz-

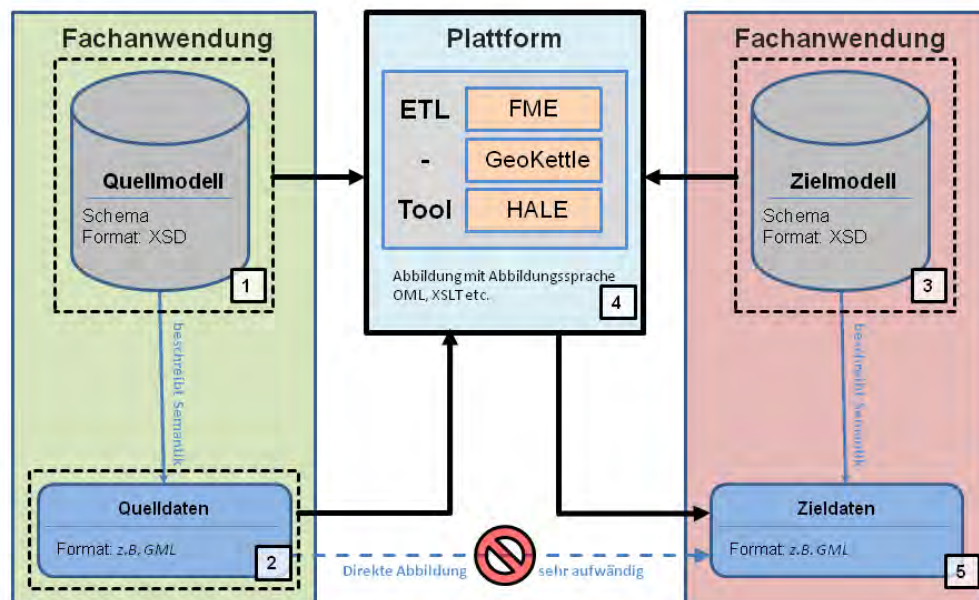


Abb. 2.10 – Prinzip der Semantischen Datenmodelltransformation mit ETL-Tools

ner und Eisenhut 2010, S.27f.]. Die Datenstruktur des originären Modells wird dabei auf ein neues Modell transferiert. Eine solche Transformation wird beispielsweise in der vorliegenden Arbeit in den Anwendungsfälle vorgenommen und findet horizontal auf gleicher Ebene der Transferformate statt (s.u.). Eine *syntaktische Transformation* beschreibt die formatbedingte Umstrukturierung der Syntax [Kutzner und Eisenhut 2010, S.27f.] und kommt beispielsweise bei der Erzeugung von GML-Anwendungsschemata aus UML-Modellen zum Einsatz. Es handelt sich dabei um eine vertikale Transformation verschiedener Ebenen (s.u.).

Zur genaueren Charakterisierung und Unterscheidung werden einige Einschränkungen bezüglich der besonderen Art der Modelltransformation getroffen:

- **Richtung der Transformation**

Die Richtung, in der eine Transformation erfolgt, gibt einerseits an, bei welchen der Schemata es sich um Quell- bzw. Zielschemata handelt. Wird die Transformation nur in eine Richtung ausgeführt - bleiben also Quelle und Ziel unverändert - findet die Transformation unidirektional statt. Soll sie in beide Richtungen möglich sein - sind also beide Modelle zugleich Quelle und Ziel - ist die Richtung als bidirektional anzugeben [Kutzner und Eisenhut 2010, S.30]. Andererseits handelt es sich bei semantischen Datenmodelltransformationen um horizontale Transformationen, da die Abbildung auf gleicher Ebene erfolgt. Sind in Quell- bzw. Zielschema hingegen unterschiedliche Ebenen betroffen, so handelt es sich um eine vertikale Transformation [Kutzner und Eisenhut 2010, S.27].

- **Offline Transformation**

Bei der offline durchgeführten Transformation wird eine Verbindung zu Netzwerken (z.B. via Internet) nicht zwingend benötigt. Es müssen dann jedoch alle benötigten Input-Daten, wie z.B. der zu transformierende Datensatz und Quell- bzw. Zielschema, lokal für den Prozessor zugänglich gespeichert sein. Im Rahmen einer GDI können Nutzer dann aber auch keine Transformationen zu einer Abfrage durchführen.

- **On-the-fly Transformation**

Durch den Terminus on-the-fly soll ausgedrückt werden, dass während der Abfrage von Datensätzen eine Transformation in das gewünschte Zielmodell erfolgen soll. Die Abfrage durch den Nutzer erfolgt demzufolge über ein Netzwerk via Bereitstellungsdiensten, die dann als Teil einer GDI implementiert sein müssen.

INSPIRE trifft keine Vorgaben dazu, dass vorhandene Datensätze im originären Datenmodell bei der Abfrage durch den Nutzer on-the-fly transformiert werden müssen. Falls diese Variante durch den Datenanbieter gewählt wird, stellt INSPIRE jedoch konkrete Anforderungen an das Leistungsvermögen des Dienstes [Europäisches Parlament und Rat 2007, S.1f.; Fichtinger 2011, S.21, S.23]:

- Geodaten sollen „auf der optimal geeigneten Ebene gespeichert, zugänglich gemacht und verwaltet werden“
- Der Dienst muss die Operationen „Get Transformation Service Metadata“ zum Abruf der Metadaten, „Transform“ zur Durchführung der Transformationen und „Link Transformation Service“ zur Weiterleitung der Transformationsanfrage der Nutzer an den Dienst unterstützen.
- „Pro Sekunde muss ein Transformationsdienst mindestens fünf Anfragen bearbeiten können.“
- „Der Dienst soll 99% der Zeit verfügbar sein.“

Je nach Umfang und Komplexität der Datenmodelle können diese Vorgaben bei der Realisierung von Netzdiensten zur Bereitstellung INSPIRE-konformer Daten problematisch werden. Laut Fichtinger (2011, S.65, verändert] plant „die Mehrheit der Datenanbieter (...) derzeit die Transformation in Form einer internen Prä-Prozessierung, bei der ein INSPIRE-konformer Sekundärdatenbestand erzeugt wird, und nicht als on-the-fly Transformationsdienst“ umzusetzen. Ein Hauptaugenmerk muss jedoch auf der Aktualität der Datensätze liegen, da dann nur eine temporäre Aktualität unmittelbar zum Zeitpunkt der offline durchgeführten Transformationen vorliegt.

Prozess der Semantischen Datenmodelltransformation

In Nissen et al. (2011, S.8ff.) wird der Prozess der semantischen Datenmodelltransformation beschrieben. Den Ausgangspunkt des Prozesses stellen die ausgewählten Quellmodelle und -daten dar, sowie das Zielmodell. Zwischen den Modellen müssen die Transformationsregeln mittels eines Schema Mappings in der sogenannten Konfigurationsphase definiert werden [Kutzner und Eisenhut 2010, S.29]. Dabei werden aus Quell- und Zielmodell korrespondierende Elemente der Schemata identifiziert und miteinander in semantische Beziehung (syn. Relation) gesetzt. Das bedeutet, dass innerhalb einer bestimmten semantischen Relation die beteiligten Quellelemente an der Bildung der beteiligten Zielelemente teilhaben. Aus den semantischen Relationen werden Transformationsregeln in Form von Funktionen abgeleitet. Somit stellt jede abgeleitete semantische Relation für sich eine Teiltransformation dar, mit der Elemente zwischen den Schemata überführt werden können. Werden viele einzelne Schemata zu einem Schema zusammengefasst, so wird dies als Schemaintegration bezeichnet.

Nissen et al. (2011, S.9f.) identifizieren vier verschiedene Typen von Teiltransformationen, der Transformation von:

1. Modellen
2. Koordinaten
3. Geometrie
4. Modellierungssprachen

Zur Definition der Teiltransformationen mit Funktionen muss durch den Dienst bzw. das ETL-Tool eine Reihe von Basiskonstrukten zur Verfügung gestellt werden. Nissen et al. (2011, S.11ff.) stellen eine umfangreiche Auswahl an Konstrukten in acht Kategorien vor, die möglicherweise bei der Transformation benötigt werden:

1. Vergleichsoperatoren
2. Arithmetische Operatoren
3. Textoperatoren
4. Datums- und Zeitoperatoren
5. Logische Operatoren
6. Identifikatoren

7. Geometrische Operatoren

8. Sonstige (z.B. administrative) Operatoren und Konstrukte

Sobald die Transformationsregeln feststehen, geht der Prozess in die Ausführungsphase über. Abbildung 2.11 zeigt einen solchen Prozess in der Ausführungsphase, wie er innerhalb einer Architektur (bzw. GDI) oder eines ETL-Tools implementiert sein kann.

Die vier Teiltransformationen werden in ETL-Tools als Teilprozesse der Modelltransformation innerhalb eines Workflows umgesetzt, können aber natürlich auch separat prozessiert werden. Nachdem die Transformationen umgesetzt wurden, muss noch eine Qualitätsprüfung erfolgen, die idealerweise im Dienst bzw. Tool implementiert ist.

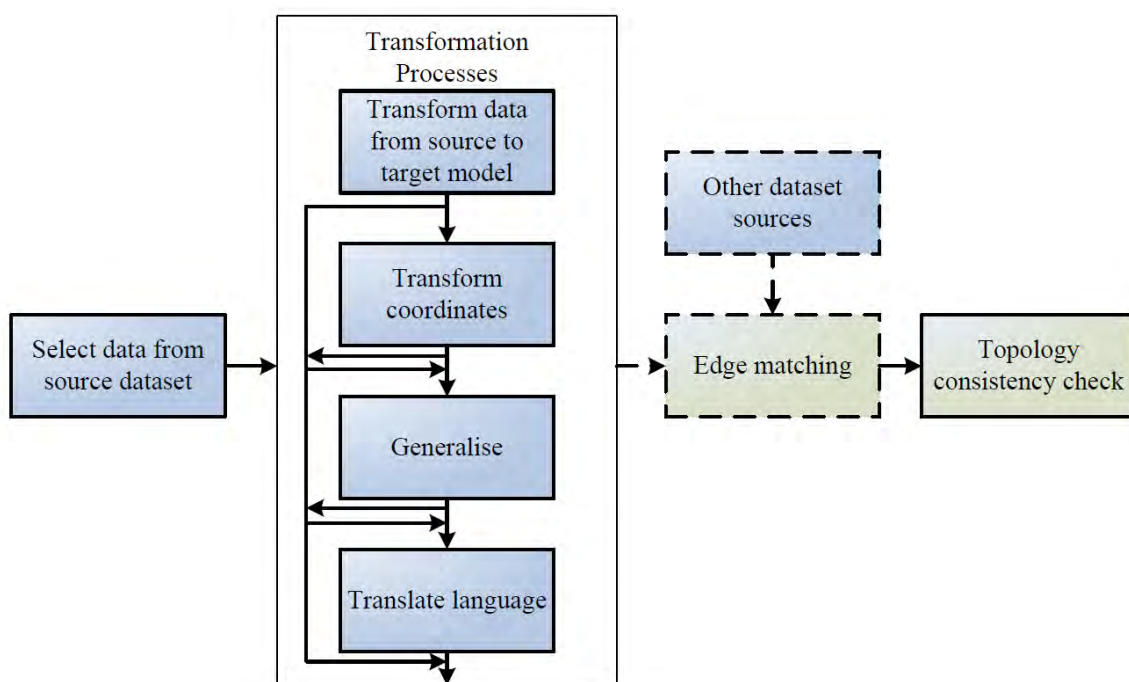


Abb. 2.11 – Prozess der Semantischen Datenmodelltransformation [Nissen et al. 2011, S.9]

2.4.2 Abbildungssprachen

Zur Abbildung bzw. Transformation zwischen Datenmodellen wird auf Abbildungssprachen (engl. Model Mapping Languages) zurückgegriffen. In Beare et al. (2010, S.17ff.) wird eine Auswahl in Frage kommender Abbildungssprachen für die Datenmodelltransformation vorgestellt und charakterisiert. Einige dieser Sprachen werden nachfolgend kurz vorgestellt:

Extensible Stylesheet Language for Transformations (XSLT)

XSLT ist eine auf den Einsatz mit XML-Dokumenten spezialisierte Transformationssprache. Sie wurde durch das W3C als offener Standard entwickelt [Fichtinger 2011, S.57; Unger 2012, S.30]. In

Beare et al. (2010, S.18) wird bei den Stärken der Sprache die weite Verbreitung der Nutzung von XML-Dokumenten und -Schemata aufgeführt, so dass die Spezialisierung von XSLT auf das XML-Paradigma sich positiv auf die Anwendung auswirkt. Darüber hinaus gelten die Stylesheets von XSLT als sehr präzise und nachvollziehbar bei der Transformation zwischen XML-Schemata. XSLT sei laut Beare et al. auch in der Lage, jede durch einen modernen Computer durchführbare Berechnung zu prozessieren. Die starke Bindung an das XML-Paradigma hat allerdings deutliche Schwächen bei der Transformation anderer Formate zur Folge.

Ontology Mapping Language (OML)

OML wurde von der Ontology Management Working Group entwickelt und im Bereich der Geoinformatik im Rahmen des Projekts HUMBOLDT zur Geospatial Ontology Mapping Language (gOML) weiterentwickelt. Mittlerweile wird die Sprache auch als EDOAL bezeichnet [Reitz 2013]. Mit der Abbildungssprache werden Korrespondenzen zwischen Ontologien von Schemata spezifiziert. Zu ihren Stärken in Bezug auf Modelltransformation von Geodaten zählt neben ihrer Spezialisierung darauf außerdem, dass es ihr möglich ist, auch komplexe Korrespondenzen zu repräsentieren, unabhängig von der Sprache, in welcher die Ontologien modelliert wurden. Allerdings befindet sich die Abbildungssprache noch in der Entwicklung und ist somit auch nicht Teil der internationalen Standardisierung [Beare et al. 2010, S.22].

Rule Interchange Format (RIF)

RIF wurde, wie auch XSLT, durch das W3C entwickelt, zum „Austausch von logischen Regeln zwischen Systemen“ [Fichtinger 2011, S.58] mit syntaktischer Repräsentation eines XML-Schemas [Beare et al. 2010, S.19]. Aufgrund des zugrunde liegenden Paradigmas ist RIF allerdings nicht geeignet Formate zu verarbeiten, die auf dem Objektorientiertem Paradigma beruhen [Kutzner und Eisenhut 2010, S.67].

Proprietäre Abbildungssprachen

Eine Reihe von Software-Produkten (darunter auch FME) haben eine eigene proprietäre Abbildungssprache zur Durchführung von Transformationen implementiert, die ausschließlich in diesen Umgebungen zur Anwendung kommen. Diese Abbildungssprachen sind somit strikt plattformspezifisch ausgelegt und nicht offen standardisiert. Welches Paradigma dabei angewandt wird, unterscheidet sich in den Produkten und ist für die Eignung im Einzelfall zu prüfen.

2.5 Objektorientiertes und Relationales Paradigma

In den vorhergehenden Abschnitten war im Bezug auf Datenmodelle und Datenmodelltransformation zumeist von einer konzeptuellen Modellierung die Rede. Tatsächlich muss in dieser Hinsicht jedoch grundsätzlich zwischen den Paradigmen unterschieden werden. Zwei dieser Paradigmen werden

nachfolgend beschrieben:

1. Objektorientierte (konzeptuelle) Modellierung

Das *Objektorientierte Paradigma* (OO-Paradigma). (Bsp.: UML, GML)

In der objektorientierten Modellierung stützt sich das Konzept auf Objektbildung, die durch Schnittstellen bearbeitbar sind (Kapselung). Zwischen den Objekten wird durch die Konzepte der Kapselung (Abgrenzung/Trennung), Vererbung und Polymorphismus indirekt eine Beziehung hergestellt. Objekte erben von einer Oberklasse Eigenschaften und Methoden. Das bedeutet, dass alle Objekte der gleichen Objektklasse dieselben vererbten Attribute und Methoden enthalten, die dann spezialisiert werden können. Durch den Polymorphismus wird zudem ermöglicht, dass Objekte Attribute und Methoden auch durch Referenzierung verwenden können, ohne diese innerhalb der Objektklasse zu definieren [Staub 2009, S.12]. Die Realisierung eines Objektes mit Programmcode wird als Instanz bezeichnet.

2. Relationale (konzeptionelle) Modellierung

Das *Relationale Paradigma*. (Bsp. Datenbanken)

Das relationale Paradigma stützt sich auf die Bildung einer Relation und wird in der Modellierung von Datenbanken eingesetzt. Zur Beschreibung der Relation werden die Inhalte in einer Tabelle gespeichert. Die Spalten der Tabelle geben die Attribute des Datenmodells wieder und die Spalten (sogenannte Tupel) fassen alle Attribute einer Zeile zu dem zusammen, was im OO-Paradigma einer Instanz entspricht. Für die Belegung der Attributwerte werden mathematische oder textuelle Repräsentationen eines definierten Wertebereichs und Datentyps herangezogen. Eines der Attribute ist in der Regel ein eindeutiger Identifikator des Tupels. Besitzt ein Tupel zudem einen Fremdschlüssel als Attribut, so verweist dieser auf den Identifikator eines Tupels einer anderen Tabelle.

Es handelt sich dabei um Sprachparadigmen einer Modellierungssprache (Abschnitt 2.1.4) in der Metamodellebene M2 (siehe Abschnitt 2.1.3). Also einer Modellierungssprache zur Beschreibung eines Realweltausschnitts anhand von Formalismen und Sprachelementen (siehe Abschnitt 2.1.2).

XML stellt ein eigenes Sprachparadigma dar: XML-Paradigma.

Bekanntlich wird ein GML-Anwendungsschema in der Formatebene durch ein XML-Schema beschrieben. Dies stellt einen gewissen Widerspruch dar, da GML objektorientiert strukturiert ist. Darüber hinaus würden sich laut Staub (2009, S.141) GML-Formate nicht eindeutig aus XML-Schemata herleiten lassen, da eine semantische Mehrdeutigkeit bestünde, die bei direkter Ableitung aus konzeptionellen Datenmodellen so nicht auftreten würde.

Kapitel 3

Beschreibungen der gewählten Fachanwendungen und der Software

3.1 Originäre Datenmodelle

Es wurden für den praktischen Teil dieser Thesis drei Datenmodelle ausgewählt, die im Folgenden vorgestellt werden.

3.1.1 Denkmalschutzgebiete in Bayern (Landesamt für Denkmalpflege)

Seit dem 19. Jhd. wird im Freistaat Bayern (ehem. Königreich Bayern) eine hoheitliche Erfassung und Katalogisierung von Bauwerken und archäologischen Fundstätten in Grund und Boden zur Denkmalpflege betrieben. Zu diesem Zweck führt das Bayerische Landesamt für Denkmalpflege (BLfD) seit dem Inkrafttreten des Bayerischen Denkmalschutzgesetzes (BayDSchG) im Jahr 1973 Denkmallisten. Diese zeigen öffentlich zugänglich alle unter das Denkmalschutzgesetz fallende und somit der staatlichen Denkmalpflege überantwortete Denkmäler auf. Seit geraumer Zeit werden die Listen in digitaler Form tagesaktuell vorgehalten und können in einem der angebotenen Viewer der Bayerischen Vermessungsverwaltungen (z.B. Bayerischer Denkmal-Atlas) online betrachtet werden [BLfD 2015]. Unter das BayDSchG fallen mitunter auch Objekte, die bislang noch nicht in der Denkmalliste erfasst sind. Tabelle 3.1 gibt einen Überblick über die wichtigsten Merkmale des Netzdienstes wieder.

Das Datenmodell unterscheidet wie das BayDSchG drei Objekttypen (FeatureTypes): Baudenkmaeler, Bauensembles und Bodendenkmaeler. Die drei FeatureTypes haben folgende Attribute gemeinsam: `uID`, `aktenNummer`, `gemeindeCode`, `gemeindeName`, `listenText`, `datum` und `geometry`.

Bei den Attributen `uID`, `hID` und `aktenNummer` handelt es sich um eindeutige Identifikatoren, die eine Unterscheidung der Einzelobjekte (Features) ermöglichen. Aber nur die FeatureTypes Baudenkmaeler und Bauensembles besitzen das Attribut `hID`. Jeder Identifikator existiert nur einmal im gesamten Datensatz, unabhängig vom FeatureType. Die Attribute `gemeindeCode` und `gemeindeName` sind die amtlichen Erkennungsschlüssel, die eine Lokalisierung bzw. Zuordnung der Features zu Gemeindegebieten ermöglichen. Mit dem Attribut `listenText` kann eine kurze individuelle Textbeschreibung

| Denkmal-Daten Bayern nach Art. 2 BayDSchG | |
|--|--|
| Bezugsquelle | nicht öffentlich zugänglich |
| Servicetyp & Version | WFS 2.0.0 |
| Ausgabeformate | GML 2.1.2, GML 3.1.1, GML 3.2.1 |
| FeatureTypeList | Baudenkmaeler, Bauensembles, Bodendenkmaeler |
| DefaultCRS | urn:ogc:def:crs:EPSG::31468 |
| Datum der Abfrage | 18. Mai 2015 |

Tab. 3.1 – Überblick über den Netzdienst des Bayerischen Landesamtes für Denkmalpflege (BLfD)

zu einem Feature angelegt werden. Dabei handelt es sich im Allgemeinen um eine bauliche bzw. architektonische Beschreibung und eine Datierung der zu schützenden Bausubstanz auf die bauliche Epoche. Das angegebene Datum entspricht dem Datum der Aufnahme des Features in die Denkmal-liste (deklatorisches Verfahren). Sämtliche Features der gelisteten Denkmalschutzgebiete werden geometrisch als 2-dimensionale Areale (Polygone) repräsentiert und sind mit amtlicher Koordinaten-Georeferenzierung¹ versehen. Die Areale (Abbildung 3.5) entsprechen im Falle von Bauwerken dem (vereinfachten) Gebäudegrundriss, in der Form, in der er ebenfalls im amtlichen Liegenschaftskataster grafisch registriert wird. Ansonsten umfasst das abbildende Areal das vermessungstechnisch abgesteckte Denkmalschutzgebiet auf Höhe der Geländeoberkante (i.d.R. Bodendenkmäler). Die geometrischen Höhen der Features werden allerdings weder geführt, noch wurden sie erfasst. Eine nicht geometrische Referenzierung der Features erfolgt mit amtlich geführten Datenvermerken, die - unterschieden je nach FeatureType - eine weitere Referenzierung respektive Objektidentifizierung ermöglichen (s.u.). Die FeatureTypes wurden durch das BLfD definiert.

Baudenkmäler (Abbildung 3.1)

FeatureType: *blfd:Baudenkmaeler*

Dieser FeatureType umfasst alle Denkmäler gemäß Art.1 Abs.2 und Art. 2 BayDSchG. Bei diesem FeatureType ist zusätzlich eine Adressierung der Features durch die Attribute *adresse* und *ortsteil* vorhanden, wobei Fälle vorkommen, bei denen ein Feature sich über mehrere Adressen oder Ortsteile erstreckt und somit die Attribute mehrere Werte aufweisen.

Bauensembles (Abbildung 3.2)

FeatureType: *blfd:Bauensembles*

Diese Form des Denkmals wird in den Art.1 Abs.3 und Art.2 des BayDSchG beschrieben. Die Features weisen außerdem das Attribut *name* als namensgebenden Bezeichner auf.

¹Die **European Petroleum Survey Group Geodesy** (EPSG) definiert EPSG-kodierte geodätische Parameter-Datensets für Georeferenzierungen.

Bodendenkmäler (Abbildung 3.3)

FeatureType: *blfd:Bodendenkmaeler*

Die Objekte, die dem FeatureType Bodendenkmaeler zuzuordnen sind, werden in den Art.1 Abs.4 und Art.2 des BayDSchG festgelegt. Eine besondere Bezeichnung dieses FeatureTypes stellt das Attribut *flurNummern* dar. Dabei handelt es sich um die Flurstücksnummer, auf dem sich das Denkmal befindet. Das Attribut wurde gewählt, da es sich oftmals um Aufgrabungen historischer Siedlungsreste handelt, die außerhalb der Gemeindeadressierungen anzutreffen sind.

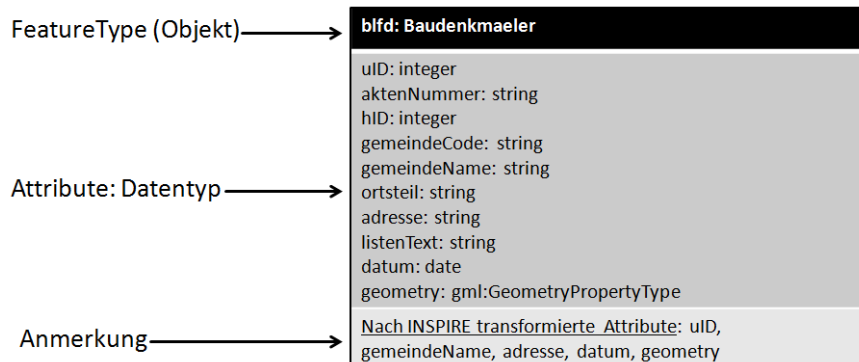


Abb. 3.1 – FeatureType Baudenkmaeler: Attribute und Anmerkungen

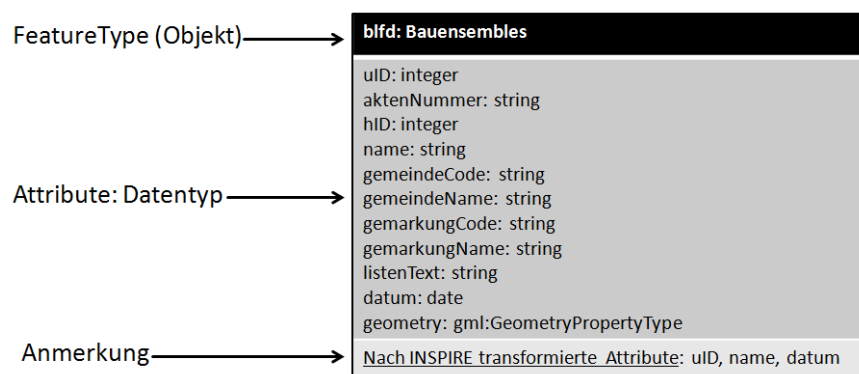


Abb. 3.2 – FeatureType Bauensembles: Attribute und Anmerkungen

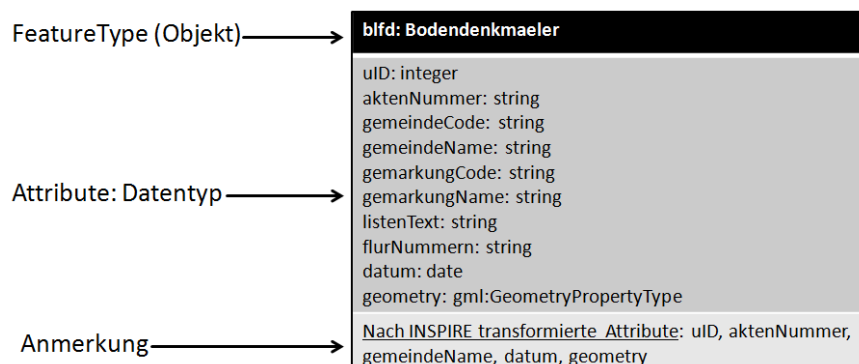


Abb. 3.3 – FeatureType Bodendenkmaeler: Attribute und Anmerkungen

Eine konzeptuelle Modellierung in Form von UML-Profilen steht für die Denkmaldaten nicht zur Verfügung. Die Datenmodelltransformation kann also ausschließlich auf Ebene der Transferformate erfolgen. Aus dem WFS wurde dazu das GML-Anwendungsschema abgerufen.

Geometrie

Die Geometrie wird mit GML-Version 3.0 durch das Geometriemodell der ISO 19107 mit gebildet [DIN 2005]. In den Denkmaldaten wird die Geometrie ausgedrückt durch das Attribut `geometry` mit dem Datentypen `GM_MultiSurface` (siehe Abbildung 3.4). Theoretisch kann es sich um jede geometrische Variante des `GM_AbstractFeatureTypes` handeln, in der Praxis wird jedoch stets `GM_MultiSurface` verwendet. Es können mehrere Einzelgeometrien Teil der gesamten Geometrie eines Features sein. Tatsächlich wurde in Stichproben kein solcher Fall in den Quelldaten entdeckt. Bei dem zugrundeliegenden Koordinatenreferenzsystem (`srsName`) handelt es sich um das in Deutschland amtliche Koordinatensystem: Gauß-Krüger Zone 4.

```

-<blfd:geometry>
  -<!--
    Inlined geometry 'BLFD_BAUDENKMAELER_2746_BLFD_GEOMETRY'
  -->
  -<gml:MultiSurface gml:id="BLFD_BAUDENKMAELER_2746_BLFD_GEOMETRY" srsName="urn:ogc:def:crs:EPSG::31468">
    -<gml:surfaceMember>
      -<gml:Polygon gml:id="GEOMETRY_6c16f8d9-2e70-43aa-a32f-e227c1b68c46" srsName="urn:ogc:def:crs:EPSG::31468">
        -<gml:exterior>
          -<gml:LinearRing>
            -<gml:posList>
              5385675.140 4561151.650 5385678.100 4561151.730 5385678.240 4561146.860 5385675.830 4561146.790 5385675.970 4561141.770 5385678.390 4561141.810 5385678.490
              4561135.500 5385687.420 4561135.640 5385687.040 4561152.290 5385689.240 4561152.340 5385689.150 4561156.430 5385686.240 4561156.370 5385686.160 4561160.090
              5385683.740 4561161.020 5385681.100 4561160.940 5385678.780 4561160.000 5385678.770 4561156.330 5385675.170 4561156.370 5385675.140 4561151.650
            </gml:posList>
          </gml:LinearRing>
        </gml:exterior>
        <gml:Polygon>
          <gml:surfaceMember>
            </gml:MultiSurface>
          </blfd:geometry>

```

Abb. 3.4 – Geometrie der FeatureTypes der Denkmaldaten des BLfD



Abb. 3.5 – Darstellung der Denkmalarten des BLfD:
Baudenkmäler, Bauensembles und Bodendenkmäler (von links)

3.1.2 Schutzgebiete in Bayern (Landesamt für Umwelt)

Dem Landesamt für Umwelt (LfU) obliegt als „zentrale Fachbehörde für Umwelt- und Naturschutz, Geologie und Wasserwirtschaft in Bayern“ [LfU 2015] die hoheitliche Aufgabe der Koordinierung, Planung und Zielsetzung für den nachhaltigen Ressourcenschutz und die Erhaltung von Pflanzen- und Tierwelt. Eine zentrale Rolle nimmt dabei die gutachterliche Bewertung von Boden, Luft, Schall und Wasser hinsichtlich von Belastungen bzw. Emissionen ein. Tabelle 3.2 gibt einen Überblick über die wichtigsten Kenndaten des Netzdienstes wieder.

| Schutzgebiets-Daten Bayern der Schutzkategorien | |
|---|---|
| Bezugsquelle | http://www.lfu.bayern.de/gdi/wfs/naturschutz/schutzgebiete? |
| Servicetyp & Version | WFS 1.1.0 |
| Ausgabeformate | GML 3.1.1 |
| FeatureTypeList | biosphaerenreservat, fauna_flora_habitat_gebiet, vogelschutzgebiet, landschaftsschutzgebiet, nationalpark, naturpark, naturschutzgebiet |
| DefaultCRS | urn:ogc:def:crs:EPSG::31468 |
| Datum der Abfrage | 18. Mai 2015 |

Tab. 3.2 – Überblick über den Netzdienst des Bayerischen Landesamtes für Umwelt (LfU)

Die Schutzgebietsdaten des LfU beinhalten sieben FeatureTypes zur Beschreibung der Schutzgebietstypen, die durch unterschiedliche Gesetze begründet sind. Durch die UNESCO² der Vereinten Nationen werden Biosphärenreservate als Schutzgebiete ausgewiesen (Abbildung 3.6). Unter der Bezeichnung NATURA 2000 wurden die Fauna-Flora-Habitat-Richtlinie und Vogelschutz-Richtlinie als Teil eines europaweiten ökologischen Verbundnetzes zusammengefasst (Abbildung 3.7). Mit diesen beiden Richtlinien wird von der EU auch eine Überwachung dieser Schutzgebiete mit Berichterstattung gefordert. Das IUCN³ beschreibt die verbleibenden vier Schutzgebiete unterschiedlicher Charakterisierung und räumlicher Ausdehnung. Alle beschriebenen FeatureTypes haben folgende Attribute gemeinsam: OBJECTID, id, name, geometry. OBJECTID ist eine laufende Zählung, welche die Features durchnummeriert. Gemeinsam mit einem vorangestellten Kürzel und getrennt durch einen Unterstrich, z.B. F35 für Landschaftsschutzgebiete und F40 für Vogelschutzgebiete, wird der eindeutige und persistente Objektidentifikator als `gml:id` gesetzt. Die vom LfU vergebene `lfu:id` ist mehrdeutig, da es sich um eine externe Nummerierung für die Schutzgebietstypen handelt, die bei manchen FeatureTypes mehrmals vorkommen kann. Das Attribut `name` beinhaltet die Namensgebung des ausgewiesenen Areals und das Attribut `geometry` grenzt das Gebiet in Form von 2-dimensionalen Polygonen ohne Höhenangabe ab.

²Die **United Nations Educational, Scientific and Cultural Organization** (UNESCO) ist eine Organisation der Vereinten Nationen zur Förderung von Bildung, Forschung und Kultur.

³Die **International Union for Conservation of Nature** (IUCN) ist eine freie Organisation zur Förderung von Natur-, Tier- und Ressourcenschutz.

Biosphärenreservate (Abbildung 3.6)

FeatureType: *lfu:biosphaerenreservat*

Biosphärenreservate werden in sich nochmal charakterisierend mit dem Attribut `zoneName` untergliedert. Ein Areal, mit einer durch das Attribut `areaSUM_ha` repräsentierten Gesamtfläche, setzt sich oftmals aus mehreren Teilarealen zusammen, deren Einzelflächen durch `areaTfl_ha` angegeben werden.

Fauna-Flora-Habitat-Gebiete und Vogelschutzgebiete (Abbildung 3.7)

FeatureTypes: *lfu:fauna_flora_habitat_gebiet* / *lfu:vogelschutzgebiet*

Die Schutzgebiete gemäß NATURA 2000 setzen sich ebenfalls aus unterteilten Arealen zusammen. Ihre Zugehörigkeit zu einem Gesamtareal wird durch Ergänzung der `id` mit einem Identifikator der Teilfläche `idTfl` erreicht. Die Flächen werden analog zu den Biosphärenreservaten angegeben.

Landschaftsschutzgebiete, Nationalparks, Naturparks und Naturschutzgebiete (Abbildung 3.8)

FeatureTypes: *lfu:landschaftsschutzgebiet* / *lfu:nationalpark* / *lfu:naturpark* / *lfu:naturschutzgebiet*

Bei diesen Schutzgebieten wird zusätzlich zur `id` noch eine Aktennummer geführt: Attribut `nr`. Eine Unterteilung der Areale erfolgt nicht, sodass lediglich deren Gesamtfläche `area_ha` angegeben wird.

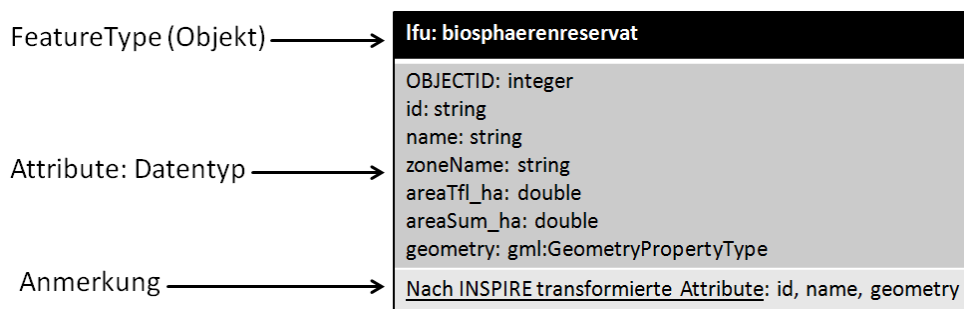


Abb. 3.6 – FeatureType biosphaerenreservat: Attribute und Anmerkungen

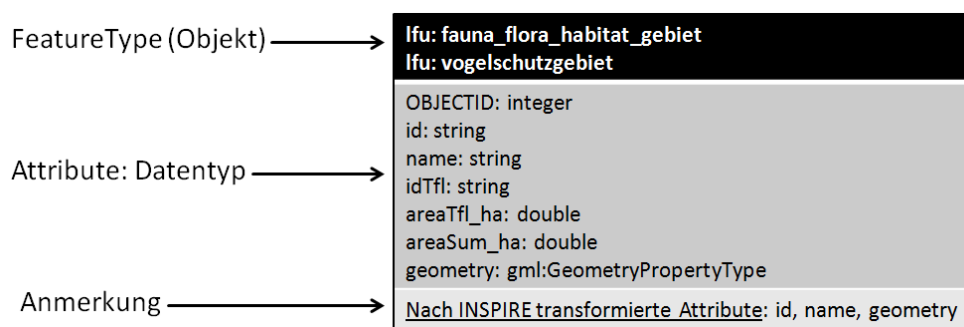


Abb. 3.7 – FeatureTypes fauna_flora_habitat_gebiet und vogelschutzgebiet: Attribute und Anmerkungen

Geometrie

Im Gegensatz zu den Denkmaldaten des BLfD ist als Elementtyp für die Geometrie der Schutzgebiete lediglich `gml:MultiSurfacePropertyType` vorgesehen. Letztlich nutzen auch beide in Instanz `gml:MultiSurfacePropertyType`. Gemäß der Eigenart der Schutzgebiete, liegen die Features als teils sehr weitläufige Gebiete vor, ohne dabei einem mathematisch stetigen Verlauf in den Gebietsgrenzen zu folgen. Dementsprechend fallen die `gml:posList`⁴ mitunter sehr umfangreich aus. Wie in Abbildung 3.9 ersichtlich, werden sehr große, eigentlich zusammenhängende Gebiete, in mehrere Features aufgeteilt (vgl. das Areal rechts unten mit der schraffierten Teilfläche). Die Koordinatenreferenz erfolgt ebenfalls in Gauß-Krüger Zone 4.

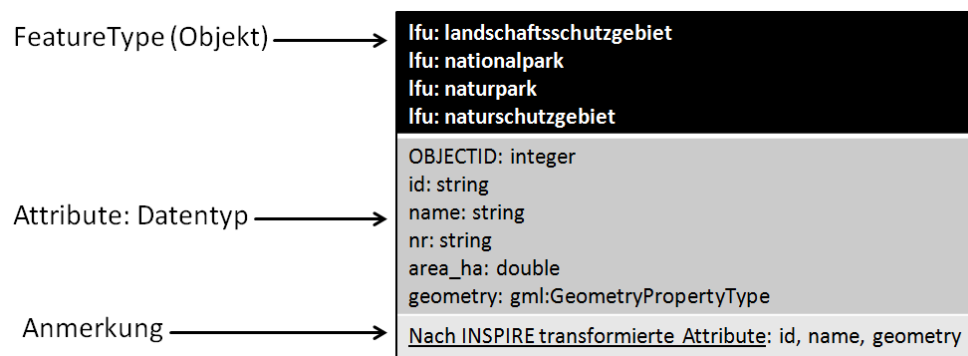


Abb. 3.8 – FeatureTypes landschaftsschutzgebiet, nationalpark, naturpark und naturschutzgebiet: Attribute und Anmerkungen

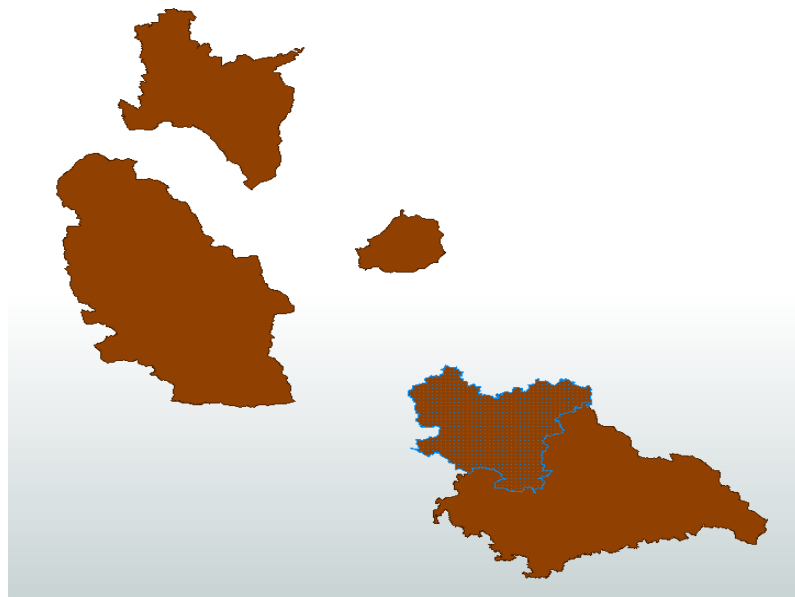


Abb. 3.9 – Darstellung der Schutzgebietsdaten des LfU am Beispiel von Naturparks

⁴Eine `gml:posList` ist ein GML-Element zur Speicherung von Koordinatenlisten innerhalb einer Geometrie.

3.1.3 AAA 6.0 CityGML 1.0

Als dritter Anwendungsfall wurde ein Gebäudemodell ausgewählt. Von der Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (AdV) wird eine Reihe von Anwendungsprofilen zur Verfügung gestellt, darunter auch ein Profil für CityGML, basierend auf CityGML 1.0: CityGML AAA 6.0. Der Datensatz des Anwendungsfalls wurde in diesem Anwendungsschema modelliert. Das Anwendungsschema wird von der AdV ausführlich im Rahmen des GeoInfoDok 6.0 beschrieben [AdV CityGML 2014].

CityGML

CityGML ist der internationaler Modellierungs- und Kodierungsstandard zur Darstellung, Datenspeicherung und -austausch von Gebäudemodellen. Die Modellierung umfasst dabei nicht nur geometrisch abgebildete architektonische Elemente, sondern außerdem auch Erscheinungsformen wie Topologie, gibt die Bedeutung (Semantik) eines Objekts und von Teilen des Objekts an und ermöglicht auch die Verknüpfung mit Referenzen und zusätzlichen Informationen [CityGML Version 1.0].

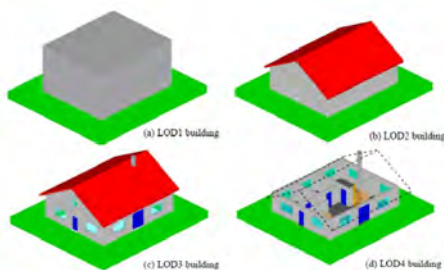


Abb. 3.10 – LoD von Gebäudemodellen
(INSPIRE DS BU 2013, S.58)

CityGML ist ein offener Standard des OGC und liegt seit 2012 in der Version *CityGML Encoding Standard 2.0* vor. Derzeit wird an Version 3.0 gearbeitet [Löwner et al. 2014]. Die Entwicklung von CityGML wurde durch das OGC gemeinsam mit der Special Interest Group 3D⁵ (SIG 3D) vorgenommen. Die Basis dafür legte GML als XML-Dialekt, das durch Erweiterung und Spezialisierung zu einem Anwendungsschema von GML für Gebäude geformt wurde. Mit CityGML ist nicht nur die Darstellung einzelner Gebäude möglich, sondern diese können in einem Verbund

zu ganzen Stadtmodellen vereint und durch topologische Objekte natürlicher (z.B. Bäume) oder anthropogener Art (andere Bauwerke wie z.B. Straßen) ergänzt werden. Prinzipiell ist sogar die Konglomeration zu ganzen Regionen und Ländern möglich [Popp 2015, S.6]. Da CityGML Teil der durch das OGC entwickelten Standards ist, lässt es sich mit den anderen Standards gut kombinieren (z.B. WFS) [CityGML 2015].

CityGML kann auch als Ausgangspunkt für weitere Anwendungen in der Geoinformatik dienen. Gegenstand der aktuellen Forschung sind Anwendungen, Gebäude- und Stadtmodelle in die Simulation diverser Umwelthanwendungen einzubinden, beispielsweise zur Ermittlung von Solarnutzungspotentialen in Kommunen, Energieverbrauchsabschätzungen, Immobilienwertermittlung oder zum Katastrophenschutzmanagement.

⁵Die **Special Interest Group 3D** (SIG 3D) ist ein Netzwerk im Bereich der Geoinformatik zur Förderung von Projekten mit 3-dimensionalem Bezug. Seit 2010 ist sie in die GDI-DE eingegliedert [SIG 3D 2015].

Nachfolgend seien einige Prinzipien aufgelistet, auf denen die Modellierung von CityGML beruht:

- **Generalisierung**

Je nach gewünschtem Detaillierungsgrad der Gebäudedarstellung stehen fünf Generalisierungsabstufungen zur Verfügung, sogenannte Level of Detail (LoD), in den Abstufungen LoD 0 bis LoD 4 (Abbildung 3.10). Die stärkste Generalisierung bzw. Abstraktion findet auf LoD 0 statt, bei dem die Objekte nur grob gestaltet dargestellt werden. LoD 1 bietet dann eine bessere Konkretisierung, indem Gebäude als sogenanntes Klötzchenmodell gebildet werden. Die Details, die in den Objekten enthalten sind, nehmen dann mit steigendem LoD zu, bis zu einem detailgetreuen Innenraummodell in LoD 4. [CityGML 2.0, S.11]

- **Flexibilität im Maßstab**

Bedingt auch durch die Generalisierung mit LoD können Stadtmodelle in verschiedensten Maßstäben und in diversen Ausdehnungen wiedergegeben werden. Dadurch lassen sich Gebäude- und Stadtmodelle an unterschiedliche Anforderungen für Anwendungen anpassen. Beispielsweise muss für eine kompetente Solarpotentialanalyse die Dachform der Gebäudes bekannt sein (LoD 2+) und die Analyse könnte von einem einzelnen Gebäude auf ein größeres Siedlungsgebiet ausgedehnt werden. [CityGML 2.0, S.11]

- **Erweiterbarkeit**

Es handelt sich um einen internationalen Standard. Es wurde die Möglichkeit geschaffen, das Schema von CityGML an eigene Anforderungen und Nutzungsmöglichkeiten anzupassen. Das Schema kann erweitert werden, indem sogenannte Application Domain Extensions (ADE) gebildet werden. Diese stellen einen „Mechanismus zur Erweiterung von CityGML um neue Eigenschaften und Objekttypen“ [CityGML 2.0, S.150ff.] dar, wie er beispielsweise genutzt werden kann, um CityGML mittels einer ADE INSPIRE-konform zu erweitern [Kolbe 2014, S.14ff.].

- **Spezialisierbarkeit**

Aus demselben Grund, warum CityGML erweiterbar ist, wird auch die Spezialisierung in Profilen ermöglicht. Ein solches Profil nimmt eine Anpassung des Schemas an eigene Bedürfnisse vor. Die Adv hat ein Profil gestaltet, um CityGML auf Bundesebene einheitlich und an die Anforderungen der Vermessungsverwaltungen anzupassen. Das Schema kann dann nicht nur erweitert werden, sondern auch reduzierend eingeschränkt und abgeändert werden (s.u.). [CityGML 2.0, S.23]

Spezialisierung im Profil AAA 6.0

Durch die Vermessungsverwaltungen der Länder werden im Liegenschaftskataster alle Gebäude „in ihrer Lage, zusammen mit weiteren Attributen“ geführt [AdV CityGML 2014, S.2], allerdings nicht in 3-dimensionaler Repräsentation. Dennoch bietet das Kataster eine hervorragende Grundlage zum flächendeckenden Aufbau von 3-dimensionalen Gebäudemodellen auf gesamtem Bundesgebiet. Mit der GeoInfoDok 6.0 wurde ein Gebäudemodell integriert, das mittlerweile neben LoD 1 auch im LoD 2 zur Verfügung steht [AdV CityGML 2014, S.2]. Das Datenausgabeformat ist CityGML Version 1.0 in den in Deutschland amtlichen Koordinatenreferenzen.

In Abbildung 3.11 ist das UML-Profil (konzeptuelle Modellierung) des Anwendungsschemas für Building (LoD 2) dargestellt [AdV CityGML 2014, S.16]. Beim Vergleich mit dem Original UML-Modell für CityGML Encoding Standard Version 1.0 [CityGML Version 1.0] fällt auf, dass die Geometrie unverändert bleibt und Einschränkungen nur im Rahmen des LoD erfolgen (z.B. FeatureType BoundarySurface) [AdV CityGML 2014, S.20f.].

Neben einigen Vereinfachungen (z.B. teilweise Reduzierung der Attribute von Klassen) wurden auch Erweiterungen vorgenommen. So wurden externe Kodierungslisten eingeführt, um Dachtypen zu registrieren (RoofType) und die Gebäudefunktion zu dokumentieren (BuildingInstallationFunction). Das zu letzterem zugehörige Attribut `function` und das Attribut `measuredHeight` wurden zu Pflichtattributen aufgewertet. Im Attribut `gml:name` des FeatureTypes Building sind nur Eigennamen der Gebäude zugelassen. Aus den Metadaten können die Verfahren und Ermittlungsmethoden entnommen werden, mit denen vermessungstechnisch erfasste Attributwerte erhoben wurden [AdV CityGML, S.21].

Was die generischen Objekte betrifft, so sind diese nicht zugelassen und generische Attribute lediglich zum Zweck, um Qualitätsangaben aus den vermessungstechnischen Ermittlungsverfahren im Format „string“ angeben zu können [AdV CityGML 2014, S.16].

Einige Module wurden aus dem Profil wurden sogar komplett entfernt (z.B. WaterBody) [AdV CityGML 2014, S.19].

Beschreibung des Datensatzes

Der zur Anwendung kommende Datensatz wurde von der AdV angelegt und bereits im Rahmen der Bachelor's Thesis von Popp (2015) verwendet, um eine semantische Datenmodelltransformation zu INSPIRE vorzunehmen.

Es handelt sich um das Modell eines Einzelgebäudes im LoD 2, das in Abbildung 3.12 aus zwei Perspektiven gezeigt wird. Das Gebäudemodell wurde in zwei Gebäudeteile (FeatureTypes: Building-Parts) unterteilt (in der linken Perspektive durch die verschiedenen Dachfarben zu unterscheiden).

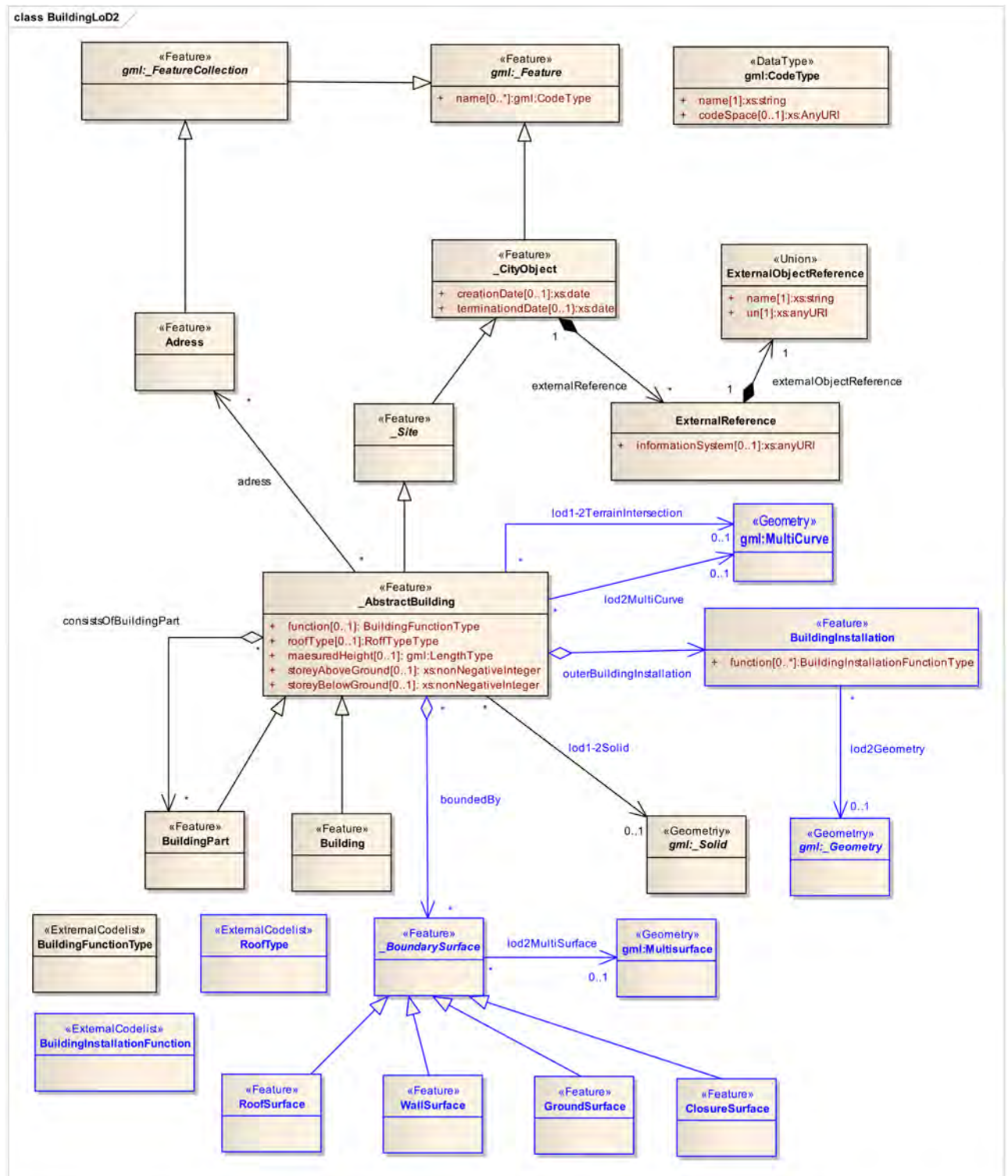


Abb. 3.11 – CityGML-Profil (GeoInfoDok 6.0) für LoD2 (Building) der AdV [AdV CityGML 2014, S.16]

Geometrie

Die Beschreibung der Geometrie beginnt mit dem Attribut `TerrainIntersection` zur Abgrenzung der Gebäudegrundfläche⁶ von der umgebenden Topologie mittels einer Verschneidung. Die eigentliche Grundfläche innerhalb der Verschneidungslinie wird von der `GroundSurface` gebildet. Die restlichen Gebäudeflächen von `WallSurface`, `RoofSurface` und `ClosureSurface`.

Die Koordinatenreferenz erfolgt in „urn:adv:crs:ETRS89_UTM32*DE\DHHN92\NH“.

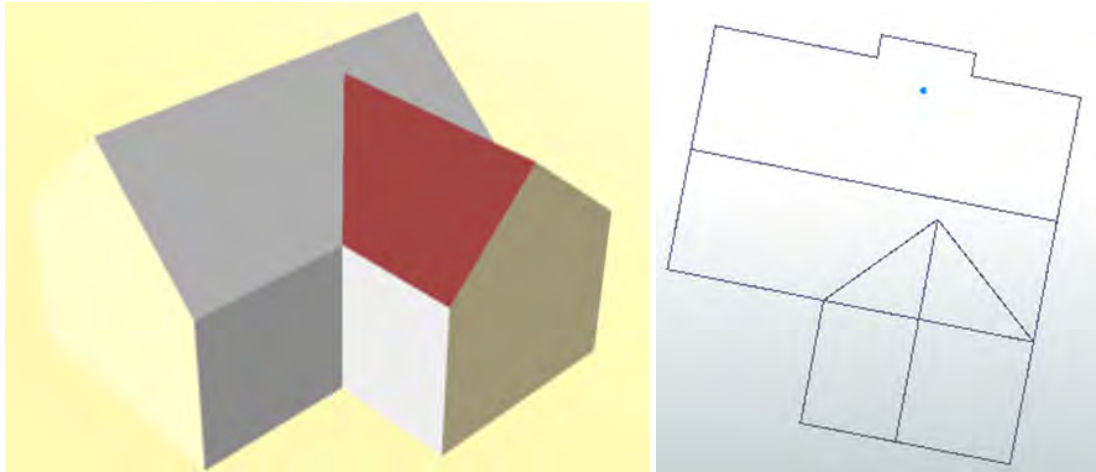


Abb. 3.12 – Darstellung des Datensatzes in 3D-Ansicht [Popp 2015, S.14] und als Draufsicht

Weitere Attribute: zusammengestellt in Tabelle 3.3.

| Attribut | Wert | Datentyp | Element von |
|--------------------------------|---------------------|--------------|-------------------------------------|
| <code>gml:id</code> | „GML Identifikator“ | string | <code>bldg:</code> |
| <code>creationDate</code> | 2013-11-05 | DateType | <code>core:</code> |
| <code>informationSystem</code> | „URL“ | URL | <code>core:externalReference</code> |
| <code>core:name</code> | „Aktenummer“ | string | <code>core:informationSystem</code> |
| Gemeindeschlüssel | 0551200 | string | <code>gen:</code> |
| function | 11_1301 | string | <code>bldg:</code> |
| DatenquelleDachhoehe | 1000 | string | <code>gen:</code> |
| DatenquelleBodenhoehe | 1400 | string | <code>gen:</code> |
| roofType | 3100 | string | <code>bldg:</code> |
| measuredHeight | 7.236 | Meter | <code>bldg:</code> |
| CountryName | Germany | string | <code>xal:</code> |
| Locality | Town | LocalityType | <code>xal:</code> |
| LocalityName | Bottrop | string | <code>xal:</code> |
| ThoroughfareNumber | 1 | string | <code>xal:</code> |
| ThouroughfareName | Maystraße | string | <code>xal:</code> |
| PostalCodeNumber | 46244 | integer | <code>xal:</code> |

Tab. 3.3 – Attribute des verwendeten Datensatzes für CityGML

⁶Die Gebäudegrundflächen können zum Beispiel dem Liegenschaftskataster entnommen werden.

3.1.4 Klassifizierung der ausgewählten Anwendungsfälle

Donaubauer (2004, S.76f.) zählt in seiner Dissertation eine Reihe von Merkmalen auf, mit denen Geodaten klassifiziert werden können. In nachfolgender Tabelle 3.4 wird anhand einiger dieser Merkmale eine Klassifikation der für diese Arbeit ausgewählten Anwendungsfälle vorgenommen.

| Merkmal | Denkmaldaten (GML 3.2.1) | Schutzgebietsdaten (GML 3.1.1) | AAA 6.0 CityGML 1.0 |
|------------------|-------------------------------------|---|--------------------------------|
| Datentyp | Vektor | Vektor | Vektor |
| Spezialisierung | Fachdaten | Fachdaten | Fachdaten |
| Thematik | Schutzgebietsdaten | Umweltdaten | Bauwerksdaten |
| Veredelung | (teil-)strukturiert | (teil-)strukturiert | strukturiert |
| Raumbezug | indirekt | indirekt | indirekt |
| Strukturierung | objektstrukturiert | objektstrukturiert | objektstrukturiert |
| Datenspeicherung | Datenbank | Datenbank | beliebig |
| Modellierung | objektorientiert | objektorientiert | objektorientiert |
| Dokumentation | nicht vorhanden | nicht vorhanden | analog und digital |

Tab. 3.4 – Klassifizierung der Anwendungsfälle gemäß Donaubauer (2004, S.76f.)

3.2 INSPIRE-Datenmodelle

Die Datenmodelle in INSPIRE werden thematisch gegliedert durch die INSPIRE Data Specifications gebildet. Zwei der ausgewählten Anwendungsfälle, Denkmal- und Schutzgebietsdaten, bilden nach Definition in INSPIRE dieselbe thematische Gruppe: Protected Sites (dt. Schutzgebiete). Damit werden sowohl bauliche und historische Bauwerke bzw. Siedlungsreste (kulturelle Schutzgebiete), sowie auch Schutzgebiete des Naturschutzes (darunter auch Kulturlandschaften) gleichgestellt. Entsprechend werden beide Modelle nach Protected Sites transformiert (Transformationsfälle I & II). Die Data Specification on Buildings hat sich aus dem CityGML Encoding Standard Version 2.0 herausgebildet und ist sehr stark an diesen angelehnt. Es ergeben sich einige Unterschiede zum Quellmodell, da es sich dabei um ein spezialisiertes Profil von CityGML Encoding Standard Version 1.0 handelt (Transformationsfall III). Abbildung 3.13 zeigt die Zusammenhänge der Modelle bei der Transformation.

Im INSPIRE Generic Conceptual Model (siehe Abschnitt 2.2.2) werden zudem Möglichkeiten aufgezeigt, die Spezifikationen innerhalb der dort aufgeführten Regeln selbstständig zu erweitern, beispielsweise durch eigene Kodierungslisten («codeList») [INSPIRE JRC 2014, S.46f., S.137f.], aber auch eigene Objekttypen. Die Erweiterungen müssen dann mit einem neuen Register aufgelistet und zugänglich gemacht, sowie in den Metadaten referenziert werden [INSPIRE JRC 2014, S.42, S.98f.]. Neben dem Stereotyp «codeList» existiert noch «enumeration» als Listentyp. Listen vom Stereotyp «enumeration» können nicht durch den Datenanbieter erweitert werden [INSPIRE JRC 2014, S.45f.]. Im INSPIRE GE (2013) wird als Standardformat zur Kodierung der Datenmodellspezifikationen das

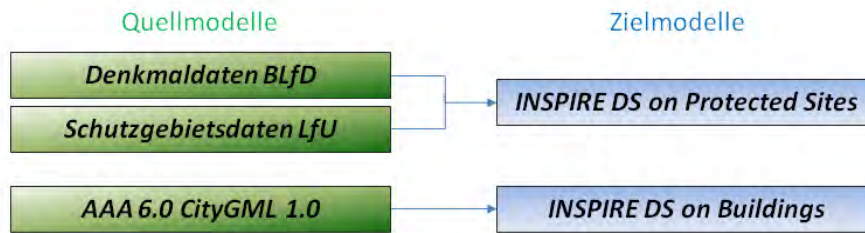


Abb. 3.13 – Darstellung der Transformationsfälle

Format GML festgelegt. Seit Einführung der Version 3.2 basiert GML auf der ISO-Reihe 19000. GML 3.2 (aktuell GML 3.2.1) selbst wurde identisch aus der Norm ISO 19136 übernommen [INSPIRE GE 2013, S.15f.; DIN 2007].

Geometrie

Mit der Festlegung des Standardformates wird auch die Geometrie des Ausgabeformates nach GML 3.2.1 bzw. ISO 19136 präzisiert, wobei die Instanzierung als `GM_Object` erfolgt. In INSPIRE DS PS (2014, S.28) werden insgesamt 21 verwendbare Koordinatenreferenzsysteme gelistet.

Regelung zur Behandlung von Attributen ohne Inhalt

INSPIRE sieht für die Handhabung mancher Attribute, die keinen Inhalt aufweisen, eine besondere Regelung vor. Hierzu wurde der Stereotyp «voidable» definiert. Zunächst wird unterschieden, aus welchem Grund (`VoidReasonValue`) kein Wert vorliegt. Das kann entweder dadurch begründet sein, dass durch den Datenanbieter keine Daten erhoben wurden bzw. kein äquivalenter Wert im Quellmodell vorgesehen war (unpopulated), oder dass einzelne Werte zu Features, aus welchen Gründen auch immer, obwohl vorgesehen nicht vorhanden sind (unknown). Zusätzlich kann noch das Attribut «nillable» mit dem Datentyp boolean geführt werden, um das Fehlen eines Wertes hervorzuheben. In der neuen Fassung der Datenspezifikationen sind mittlerweile drei Werte zur Angabe des `VoidReasonValue` vorgesehen, jedoch werden in der vorliegenden Arbeit nur die zwei Werte der hier verwendeten Fassungen genutzt.

Eine wichtige Rolle spielt dabei die Kardinalität⁷ der Attribute. Beträgt diese 0, so muss das Attribut bei der Instanzierung nicht implementiert werden. Wird hingegen ein anderer Wert geführt (1,*), müssen auch «voidable»-Attribute ohne Wert, mit Angabe des Grundes berücksichtigt werden.

3.2.1 INSPIRE Data Specification on Protected Sites

Die Data Specification on Protected Sites betrifft das Geodaten-Thema in Anhang I Punkt 9. der INSPIRE-Richtlinie. Die Definition in der Richtlinie lautet wie folgt [Europäisches Parlament und Rat 2007, Anhang I.9; INSPIRE ATS 2008, S.44]:

⁷Die Kardinalität (syn. Multiplizität) trifft eine Aussage über die mögliche Häufigkeit des Auftretens eines Attributs.

„Gebiete, die im Rahmen des internationalen und des gemeinschaftlichen Rechts sowie des Rechts der Mitgliedsstaaten ausgewiesen sind oder verwaltet werden, um spezifische Erhaltungsziele zu erreichen.“

Bislang wurde das Datenmodell noch in eine normale und vereinfachte (engl. simple) Version unterschieden. In Zukunft wird es nur noch auf eine zusammengefasste Version beschränkt sein. Das Datenmodell [INSPIRE DS PS 2014, S.18] wurde durch die Vereinigung der Landesdenkmalpfleger in Kooperation mit der GDI-BY zu einem deutschen Profil erweitert. Hinzugefügt wurde eine Kodierungsliste zur Klassifizierung des Schutzgebietstypen in deutscher Sprache («codeList» GermanMonumentsRecordDesignationValue). Des Weiteren umfasst die Profilerweiterung den FeatureType GermanMonument mit dem Attribut designationLegalDefinition und Attributwert aus den Listenelementen der «codeList» StateLegalDefinitionsDesignationRecordValue. Allerdings liegen die Erweiterungen in noch nicht endgültig beschlossener Fassung vor. Das Klassendiagramm der Datenspezifikation mit deutscher Erweiterung ist den Anhängen C.1 und C.2 zu entnehmen. Durch die Spezifikation wird auch die Bereitstellung der Daten in Form von Layern geregelt. Die Benennung des Layers lautet „PS.ProtectedSites“ [INSPIRE DS PS 2014, S.49]. Das zugrundeliegende XML-Schema (XSD) ist in folgender URI hinterlegt [INSPIRE DS PS 2014, S.48]:
<http://inspire.ec.europa.eu/schemas/ps/3.0/ProtectedSites.xsd>

Im Gegensatz zu den Quellmodellen erfolgt die Unterscheidung der Schutzgebietstypen nicht mehr durch unterschiedliche FeatureTypes, sondern durch klassifizierende Attributwerte aus den Kodierungslisten. Das Herzstück des Modells stellt somit nur noch eine Objektklasse: FeatureType ProtectedSite. Tabelle 3.5 zeigt eine Übersicht über die in der Spezifikation verwendeten Typen und weist ihnen eine Nummerierung zur späteren Verwendung in dieser Thesis zu.

| Stereotyp | Type |
|---------------|--|
| «featureType» | ProtectedSite |
| «codeList» | DesignationSchemeValue, DesignationValue, IUCNDesignationValue, NationalMonumentsRecordDesignationValue, Natura2000DesignationValue, RamsarDesignationValue, UNESCOManAndBiosphereProgrammeDesignationValue, UNESCOWorldHeritageDesignationValue |
| «dataType» | DesignationValue |
| «enumeration» | ProtectionClassificationValue |

Tab. 3.5 – Liste der in INSPIRE DS PS (2014) verwendeten Typen

3.2.2 INSPIRE Data Specification on Buildings

Die Data Specification on Buildings betrifft das Geodaten-Thema in Anhang III Punkt 2. der INSPIRE-Richtlinie. Dort wird das Thema folgendermaßen definiert [Europäisches Parlament und Rat 2007,

Anhang III.2; INSPIRE ATS 2008, S.61]:

„Geografischer Standort von Gebäuden.“

Selbstverständlich ist die Spezifikation deutlich umfangreicher, als es die Definition vermuten lässt. Der Aufbau des Datenmodells erfolgt modular und hierarchisch, um eine Anpassung des Modells an eigene Anforderungen in Umfang und Komplexität vornehmen zu können. Wie Abbildung 3.14 entnommen werden kann, werden die modular-hierarchischen Unterscheidungen in den Merkmalen der Geometrie (2-dimensional, 3-dimensional) und Semantik (Core, Extended) vorgenommen. Dies führt zu vier anwendbaren Profilen⁸: BuildingsCore2D, BuildingsExtended2D, BuildingsCore3D und BuildingsExtended3D. In der Variante Core (basiert auf Base) kommen die FeatureTypes Building und BuildingPart zur Anwendung, in der erweiterten Variante Extended (basiert auf Extended Base) zusätzlich die FeatureTypes BuildingUnit, Installation und OtherConstruction. Sollen nicht nur Gebäude modelliert werden, sondern auch andere Bauwerke, wie beispielsweise Brücken (OtherConstruction), muss auf das umfangreichere Extended-Profil zurückgegriffen werden. Dafür lassen sich damit alle Bauwerke eines Stadtmodells im einheitlichen Datenmodell gestalten. Als Zielmodell für den Anwendungsfall CityGML wurde das Profil BuildingsCore3DLoD2 ausgewählt.

In Anhang C.3 ist das Klassendiagramm zur Datenspezifikation im Fall BuildingsCore3D dargestellt.

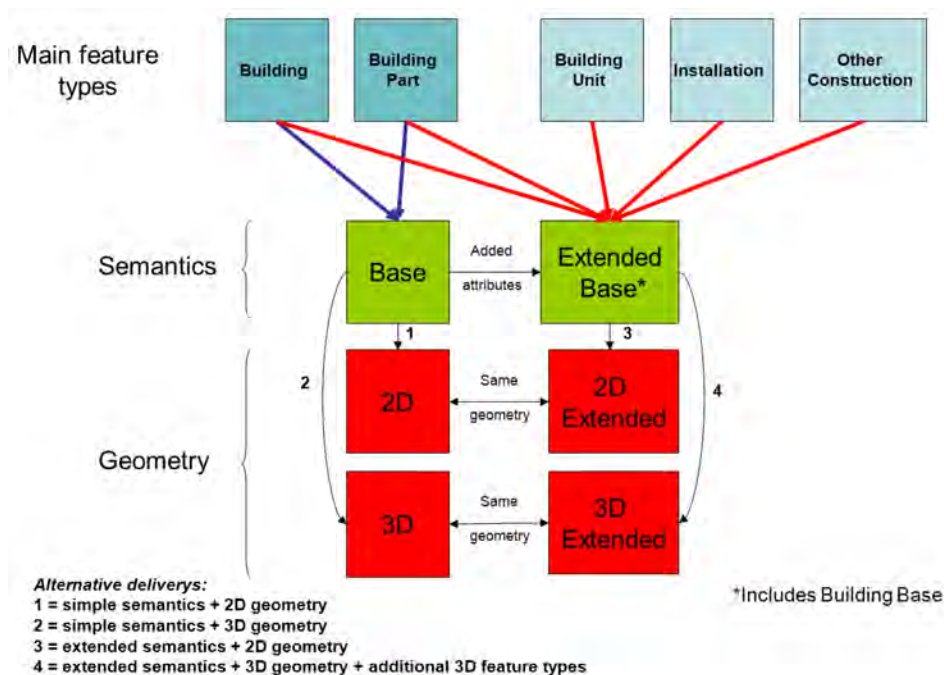


Abb. 3.14 – Struktur der Anwendungsschemata für INSPIRE Buildings [INSPIRE DS BU 2013, S.6]

Verfügt ein Objekt über BuildingParts (dt. Gebäudeteile), muss die Geometrie durch diesen FeatureType wiedergegeben werden (Bedingung: MandatoryGeometry), falls nicht muss sie in Building

⁸Die Grundlage für CityGML gemäß INSPIRE bildet GML in der Version 3.2.1.

enthalten sein (Bedingung: GeometryWhenNoParts). BuildingParts ist über eine Assoziation⁹ (Komposition¹⁰) der Klasse Building zugeordnet. Das bedeutet, dass Building zwingend existieren muss und BuildingParts, welche Building in Gebäudeteile unterteilen, optional existieren können [INSPIRE DS BU 2014, S.62].

Geometrie

Die Geometrie wird im Profil BuildingCore3D (LoD2) von dataType BuildingGeometry3DLoD2 gebildet. Bei den möglichen Geometrietypen handelt es sich dann um GM_MultiSurface und GM_Solid, wie auch im Profil AAA 6.0 CityGML 1.0. Beide beschreiben die äußere Gebäudehülle, unterscheiden sich jedoch darin, dass bei GM_MultiSurface die Grundfläche fehlt. Als horizontale Positionsreferenz für den 3-dimensionalen Volumenkörper kann die Gebäudegrundfläche (BuildingGeometry2D) dienen, welche einem 2-dimensionalen Profil entstammt, muss aber nicht («voidable», aber Multipizität 0..*). Punktreferenzierungen sind in dieser Geometrie ausgeschlossen [INSPIRE DS BU 2014, S.56]. Neben dem Stereotyp «voidable» wird noch ein zweiter Stereotyp «lifeCycleInfo» eingeführt, zum Zweck der Dokumentation des Lebenszyklus und der Version der Daten.

3.3 Beschreibungen der Transformationswerkzeuge

In den nachfolgenden Abschnitten 3.3.1 bis 3.3.3 werden die ausgewählten Datenmodelltransformatiionswerkzeuge (Tabelle 3.6) vorgestellt. Alle Tools wurden, zum Zeitpunkt mit Beginn der Bearbeitungszeit dieser Thesis, in der aktuellsten Version angewandt.

| Name | Abk. | Abbildungssprache | Lizenz |
|-----------------------------|------|-------------------|------------|
| Feature Manipulation Engine | FME | proprietär | proprietär |
| GeoKettle | - | - | GNU 2.0 |
| HUMBOLDT Alignment Editor | HALE | EDOAL/gOML | FOSS |

Tab. 3.6 – Übersicht über verwendete Software für semantische Transformationen

3.3.1 Feature Manipulation Engine

Die proprietäre Softwarefamilie **Feature Manipulation Engine** (FME) wird seit 1996 von der kanadischen Firma Safe Software Inc. entwickelt. Dazu gehören die Softwareprodukte FME Desktop, FME Server und FME Cloud. Das eigentliche ETL-Tool der Familie zur Verarbeitung und Transformation von Geodaten stellt FME Desktop dar (Tabelle 3.7).

⁹Eine Assoziation beschreibt in einem UML-Diagramm die Beziehung zwischen Objekten.

¹⁰Eine Komposition ist in UML eine Sonderform der Assoziation, die Zugehörigkeiten von Objekten zu anderen beschreibt.

| FME Desktop 2015 | |
|---------------------------------|---|
| Edition | FME Oracle Edition |
| Version | FME 2015.0 (WIN 64) |
| Unterstützte Plattformen | Windows (64-bit, 32-bit), Mac, Linux, virtualisierte Plattformen |
| Systemvoraussetzungen | Prozessor: Intel® Pentium® 4 (Minimum) Arbeitsspeicher: 4 GB RAM (32-bit), 8 GB RAM (64-bit) Festplattenspeicherplatz: ca. 2 GB (ohne Datensätze) |

Tab. 3.7 – FME Desktop, Datentabelle der genutzten Edition [Safe Inc. 2015]

Das Programm teilt sich in FME Workbench zur Bearbeitung der Aufgaben und FME Data Inspector zur Visualisierung von Datensätzen. Beide Applikationen werden separat als EXE-Dateien ausgeführt. FME Workbench stellt dem Anwender eine grafische Benutzeroberfläche /-schnittstelle (engl. Graphical User Interface) zur Verfügung.

FME Desktop Graphical User Interface (GUI)

In Abbildung 3.15 ist die Benutzerschnittstelle zu sehen. Der Aufbau der GUI in der Abbildung setzt sich aus (1) Menü- sowie (2) Werkzeugleiste, einer (3) grafischen Arbeitsoberfläche (engl. Panel), einer (4) Informationsbox, einem (5) Navigationsmenü und einem (6) Auswahlmenü für Funktionen als wichtigste Teilbereiche zusammen. Es ist möglich, diese und weitere Anzeigefenster nach eigenen Bedürfnissen anzuordnen.

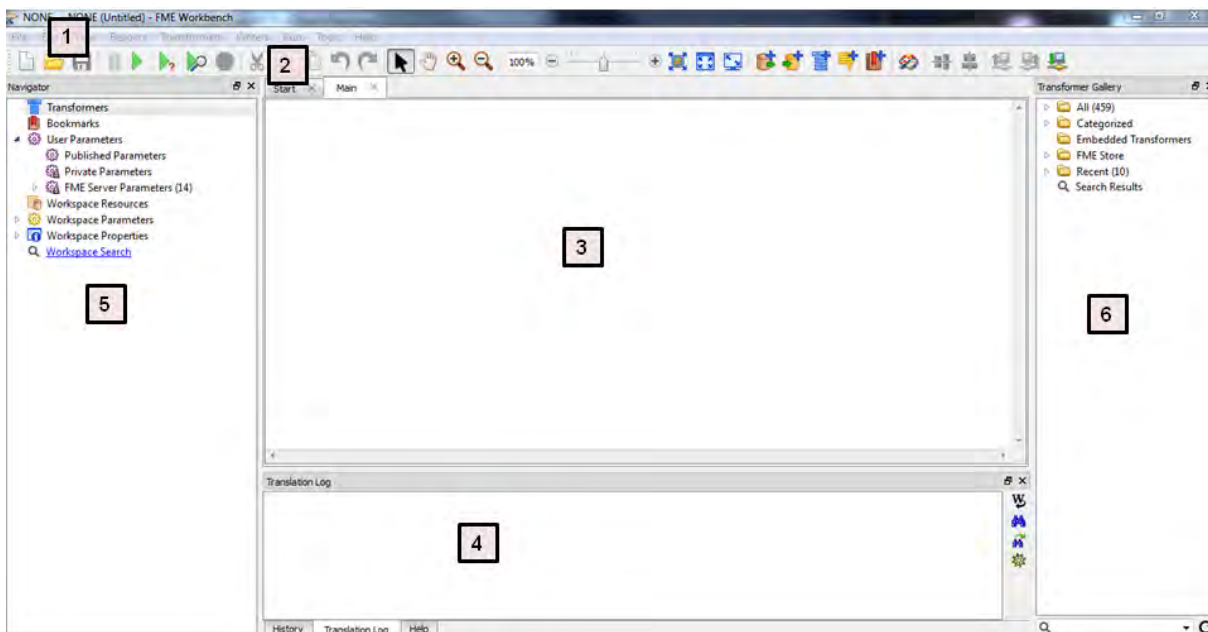


Abb. 3.15 – FME: Oberfläche der Benutzerschnittstelle (GUI)

Konfiguration der Transformation mit FME

Auf der Arbeitsfläche (3) können die Aufgaben, die zur Durchführung geplant sind, arrangiert werden. Hierzu wird mit grafischen Elementen eine sequentielle Ablaufsteuerung initiiert. Den ersten Schritt stellt das Einlesen von Datensätzen oder Schemata dar, wozu eine Einlesefunktion (Reader) aus der Werkzeugleiste (2) benötigt wird. Es ist auch möglich mehrere Datensätze oder Schemata gleichzeitig einzuladen und zu verarbeiten. Aus dem Auswahlménü (6) kann aus über 400 Funktionen¹¹ zur Bearbeitung der Daten gewählt werden. Die Steuerung der zeitlichen Abfolge der anzuwendenden Funktionen wird durch Platzierung und in-Beziehung-setzen der grafischen Funktionsrepräsentationen (Transformer) nach Rangfolge erreicht. Zum Abschluss werden die Daten mit einer Funktion (Writer) als neuer Datensatz geschrieben. Ein fertiger Arbeitsablauf wird Workspace genannt. Es lassen sich auch mehrere Workspaces ineinander einbinden.

Unterstützte Datenformate

FME kann das von INSPIRE geforderte Format GML 3.2.1 verarbeiten. Da es sich bei Safe Inc. um den Branchenführer handelt, werden Weiterentwicklungen von Formaten in der Regel zügig in die Software integriert. Im Folgenden werden die wichtigsten unterstützten Datenformate aufgelistet, unterschieden danach, ob sie von der Software gelesen oder geschrieben werden. Die meisten Formate werden von FME auch in älteren Versionen unterstützt.

Unterstützte Eingangsdatenformate:

u.a. CityGML 2.0, Comma-separated values (CSV), Extensible Markup Language (XML), Extensible Schema Definition (XSD), Geography Markup Language 3.2.1 (GML), INSPIRE GML, Keyhole Markup Language 2.2 (KML), Microsoft Excel (XLSX), Shapefile (SHP), Textfile (TXT), Web Feature Service Input (WFS), viele gängige Datenbankformate (u.a. Oracle, PostgreSQL/PostGIS, MySQL) [FME Workbench Help 2015].

Unterstützte Ausgangsdatenformate:

u.a. CityGML 2.0, Comma-separated values (CSV), Extensible Markup Language (XML), Geography Markup Language 3.2.1 (GML), INSPIRE GML, Keyhole Markup Language 2.2 (KML), Microsoft Excel (XLSX), Shapefile (SHP), Textfile (TXT), viele gängige Datenbankformate (u.a. Oracle, PostgreSQL/PostGIS, MySQL) [FME Workbench Help 2015].

3.3.2 GeoKettle

Das Open-Source ETL-Tool GeoKettle (Tabelle 3.8) wurde in den Jahren 2001 bis 2008 von der Pentaho Corporation entwickelt (zunächst unter der Bezeichnung Pentaho Data Integration). Zwischenzeitlich gingen die Urheberrechte von 2007 bis 2009 auf die GeoSOA Research Group (Laval University) über und liegen nun seit 2009 bei der Spatialytics solutions Inc., mit Sitz in Québec, Kanada

¹¹FME stellt jede Funktion in einer als Transformer bezeichneten Repräsentation zur Verfügung, über die auch die Variablen einer Funktion manipuliert werden können.

[Spatialytics 2015]. Ursprünglich bestand das Programm aus vier einzelnen Konsolenanwendungen (spoon, pan, kitchen, carte), denen getrennte Aufgabenbereiche zukamen (z.B. spoon: Anlegen der Transformationsaufgaben). Das Programm wurde auf Basis der Programmiersprache Java entwickelt und bezieht seine Funktionalität aus diversen Java Bibliotheken: GeoTools, deegree und OGR. Des Weiteren ist die Funktionalität via Plug-ins¹² erweiterbar [Vautard 2013].

Die Software steht unter der Lizenz GNU¹³ 2.0 zur freien Verfügung.

| GeoKettle's GUI | |
|---------------------------------|--|
| Version | 2.5 (WIN 32) |
| Unterstützte Plattformen | Windows (32-bit), Mac, Linux |
| Systemvoraussetzungen | Prozessor: unbekannt Arbeitsspeicher: unbekannt Festplattenspeicherplatz: ca. 160 MB (ohne Datensätze) |
| Weitere Anforderungen | Java Runtime Environment (JRE) Version 5 (oder neuer) |

Tab. 3.8 – GeoKettle, Datentabelle der genutzten Edition [Spatialytics 2015]

GeoKettle Graphical User Interface

Mittlerweile steht für das Programm eine grafische Benutzeroberfläche zur Verfügung, die in Abbildung 3.16 zu sehen ist. Der Aufbau setzt sich typischerweise aus einer (1) Menü- und (2) Werkzeugleiste zur Einstellung der Software und der Arbeitsprozesse, einer (3) grafischen Arbeitsfläche zur Gestaltung der Transformationen und einem (4) Auswahlmenü der Funktionen zusammen.

Konfiguration der Transformation mit GeoKettle

Die Konfiguration der Transformationen in GeoKettle erfolgt ähnlich wie in FME. Auf der Arbeitsfläche (3) werden grafische Funktionsrepräsentationen (in GeoKettle Steps genannt) mit verbindenden Assoziationen (Hops) in der Rangfolge angeordnet, in der sie abgearbeitet werden sollen. Die Funktionsbibliothek umfasst knapp über 100 einzelne Funktionen in 18 verschiedenen Rubriken.

Unterstützte Datenformate

GeoKettle kann das von INSPIRE geforderte Format GML 3.2.1 **nicht** verarbeiten. Es ist auch nicht abzusehen, wann ein Update erscheinen wird, das die Handhabung von GML 3.2.1 ermöglicht. Die aktuelle Version wurde im Februar 2013 veröffentlicht, während die Publikation von GML 3.2.1 im Jahr 2007 erfolgte. Nachfolgend werden die wichtigsten von GeoKettle unterstützten Datenformate gelistet, die gelesen oder geschrieben werden können.

¹²Ein Plug-in ist die Erweiterung von Softwarefunktionalität mit gekapseltem Programmcode.

¹³Die GNU **G**eneral **P**ublic **L**icense (GPL) ist eine Lizenz für freie Software (Open Source).

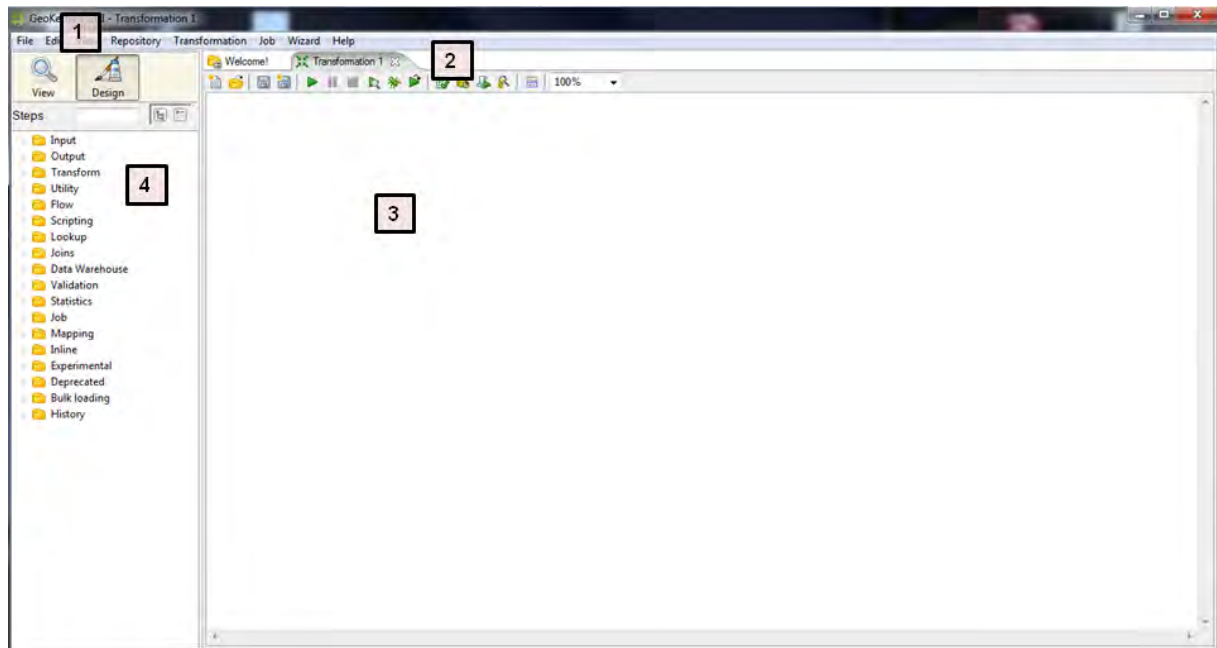


Abb. 3.16 – GeoKettle: Oberfläche der Benutzerschnittstelle

Unterstützte Eingangsdatenformate:

u.a. Comma-separated values (CSV), Extensible Markup Language (XML), Extensible Schema Definition (XSD), Geography Markup Language 3.1.1 (GML), Keyhole Markup Language 2.2 (KML), Microsoft Excel (XLSX), Shapefile (SHP), Textfile (TXT), Web Feature Service Input (WFS), diverse Datenbankformate (u.a. Oracle, PostgreSQL/PostGIS, MySQL) [GeoKettle Dokumentation 2015].

Unterstützte Ausgangsdatenformate:

u.a. Extensible Markup Language (XML), Geography Markup Language 3.1.1 (GML), Keyhole Markup Language 2.2 (KML), Microsoft Excel (XLSX), Shapefile (SHP), Textfile (TXT), diverse Datenbankformate (u.a. Oracle, PostgreSQL/PostGIS, MySQL) [GeoKettle Dokumentation 2015].

3.3.3 HUMBOLDT Alignment Editor

Im Jahr 2006 wurde das HUMBOLDT Forschungsprojekt ins Leben gerufen, um im Kontext von INSPIRE die Datenharmonisierungsprozesse zur Schaffung einer einheitlichen europäischen Geodateninfrastruktur zu unterstützen [HUMBOLDT project 2010]. Unter Federführung des Fraunhofer Instituts für Geographische Datenverarbeitung (IGD) arbeiteten während des Projekts insgesamt 28 Organisationen aus 14 europäischen Nationen zusammen. Mit Abschluss des Projektes in 2011 konnte das HUMBOLDT Framework präsentiert werden [Fichtinger et al. 2010]. Das Herzstück des Projekts war die Implementierung eines Tools zur Durchführung von Transformationen, dem HUMBOLDT Alignment Editor (HALE), in das zwischenzeitlich auch der für das Framework konzipierte Conceptual

Schema Transformer integriert wurde (Tabelle 3.9). HALE steht als Open-Source-Software (FOSS¹⁴) zur Verfügung. Die verfügbare Dokumentation bezieht sich allerdings auf eine ältere Version [Reitz und Templer 2011].

| HUMBOLDT Alignment Editor (HALE) | |
|---|--|
| Version | 2.9.2 (WIN 64) |
| Unterstützte Plattformen | Windows (64-bit, 32-bit), Mac, Linux |
| Systemvoraussetzungen | Prozessor: unbekannt Arbeitsspeicher: unbekannt Festplattenspeicherplatz: ca. 300 MB (ohne Datensätze) |

Tab. 3.9 – HALE, Datentabelle der genutzten Edition [ESDIN 2015]

HALE Guided User Interface

Um dem Nutzer den Umgang mit HALE zu erleichtern wurde eine grafische Benutzerschnittstelle für den Editor gestaltet. In Abbildung 3.17 ist das GUI in der getesteten Version für Microsoft Windows zu sehen. Neben der typischen (8) Menüleiste und den ein- bzw. ausblendbaren Toolbars für Shortcuts handelt es sich bei den übrigen grafischen Anzeigeelementen um Panels (1-6), die innerhalb des Programmfensters frei arrangiert werden können. Die Panels dienen in erster Linie der Anzeige von Strukturen und Elementen, erlauben aber auch die Durchführung von Aktionen durch Auswahl (z.B. markieren durch anklicken), oder der Betätigung von Schaltflächen.

Im (1) Schema Explorer werden die geladenen Anwendungsschemata - (genau) ein Quellschema und (genau) ein Zielschema - in hierarchischer Objektstruktur des jeweiligen Schemas aufgeschlüsselt. Das Panel für das (2) Alignment zeigt die zugewiesenen Relationen und Funktionen zwischen den FeatureTypes und Attributen der Schemata auf. Um im Zielschema die gewünschten Attributwerte zu erhalten, kann auf die (5) Funktionsbibliothek zurückgegriffen werden. Wurden bereits Quelldaten geladen, können die Eigenschaften der Features in Instanz (u.a. Attributwerte der Instanzen) betrachtet werden (4). Noch während die Abbildungsregeln umgesetzt werden, besteht mit hinzugeladenen Quelldaten die Möglichkeit, mit einer aktivierten Transformation zur Laufzeit die transformierten Zieldaten zu betrachten (6). Die Ausführung der Transformation wird durch den integrierten *Conceptual Schema Transformer* (CST) mit der Abbildungssprache EDOAL/gOML durchgeführt (7).

Konfiguration der Transformation mit HALE

Das Arrangement erfolgt direkt zwischen den Elementen der geladenen Schemata. Hierzu werden im Schema Explorer (Abbildung 3.18) die jeweiligen korrespondierenden Attribute beider Schemata ausgewählt und über Funktionen assoziiert. In HALE wird dieser Vorgang als Alignment bezeichnet und im Alignment Editor (Abbildung 3.19) des Tools abgebildet. Eine zeitliche Abfolge zur Abarbeitung der Funktionen ist nicht gegeben. Nach Fertigstellung des Alignments wird der Anwender bei der Durchführung der Transformation, bis zu deren Fertigstellung, durch eine Abfolge von Einstellungsmenüs geleitet.

¹⁴Eine **Free/Libre Open Source Software** (FOSS) ist eine freie Software (Open Source).

Unterstützte Datenformate

HALE kann das von INSPIRE geforderte Format GML 3.2.1 verarbeiten. Da die Entwicklung von HALE im Kontext von HUMBOLDT stattfand, sind die handhabbaren Formate speziell auf Konformität zu INSPIRE hin ausgerichtet. Folglich ist auch zu vermuten, dass Updates mit Bezug auf die Formate, stark an INSPIRE gebunden bleiben. Es folgen die Auflistungen der unterstützten Formate für den In- und Output des Tools.

Unterstützte Eingangsdatenformate:

u.a. Comma-separated values (CSV), Extensible Markup Language (XML), Geography Markup Language 3.2.1 (GML), Microsoft Excel (XLSX), Shapefile (SHP), Web Feature Service Input (WFS), Datenbankformate (PostgreSQL, PostGIS).

Unterstützte Ausgangsdatenformate:

u.a. Comma-separated values (CSV), Extensible Markup Language (XML), Extensible Stylesheet Language for Transformations transformation (XSLT transformation), Extensible Stylesheet Language for Transformations transformation CityGML (XSLT transformation CityGML), Geography Markup Language 3.2.1 (GML), HALE Schema Definition (HSD, HALESHEMA), INSPIRE GML, Microsoft Excel (XLSX), Datenbankformate (PostgreSQL, , PostGIS).

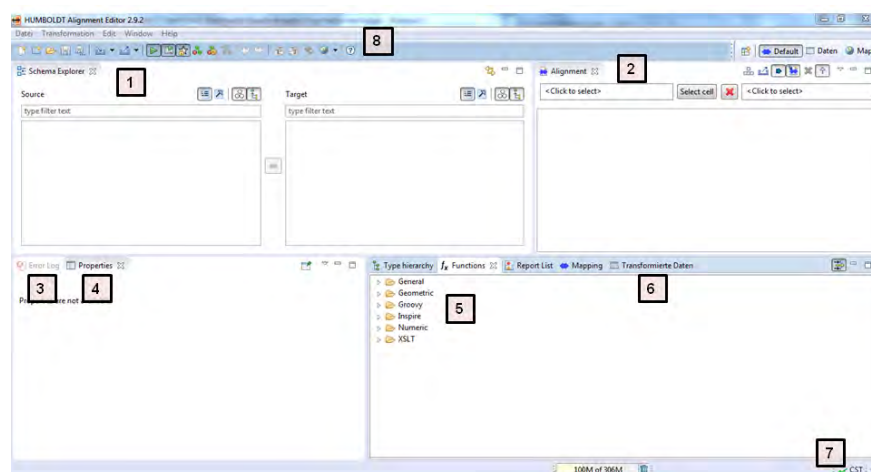


Abb. 3.17 – HALE: Oberfläche der Benutzerschnittstelle

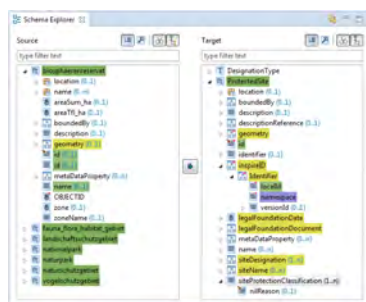


Abb. 3.18 – HALE panel - Schema Explorer



Abb. 3.19 – HALE panel - Alingment Editor

Kapitel 4

Methodisches Vorgehen

In diesem Kapitel wird das Methodische Vorgehen erläutert, mit dem die Ergebnisse der Transformationen generiert und die Bewertung der Datenmodelltransformationswerkzeuge durchgeführt wurden.

4.1 Ausgangssituation

Die Transformationen sollen auf Ebene der Transferformate durchgeführt werden. Durch das LfU werden Netzdienste (u.a. WFS) betrieben, mittels denen die Datensätze im Transferformat GML abgerufen werden können. Über den gleichen Service wird auch das Schema (XSD) zu den Datensätzen veröffentlicht. Zum Datensatz des BLfD wurde für diese Arbeit der Zugang zu einem WFS ermöglicht, welcher ansonsten nur für interne Zwecke des BLfD genutzt wird. Die verwendeten Daten wurden ebenfalls im Transferformat GML geordnet bzw. das zugehörige Schema im Format XSD. Allerdings ist, wie bereits erwähnt, eine konzeptuelle Modellierung in Form von UML-Profilen in beiden Fällen nicht vorhanden, so dass weitere Informationen über die Datenmodelle nur über die Abfragen *GetCapabilities*¹ und *DescribeFeatureType*² gewonnen werden konnten.

Das LfU stellt durch seinen Netzdienst die Daten im Format GML 3.1.1 zur Verfügung, während die Daten des BLfD in Version 3.2.1 abgerufen werden konnten.

Für den Anwendungsfall AAA 6.0 CityGML 1.0 wird ein nicht öffentlich zugänglicher Testdatensatz der AdV verwendet. Das zugrundeliegende Datenmodell wurde bereits zuvor referenziert (siehe Abschnitt 3.1.3)

Somit ergeben sich drei durchzuführende Datenmodelltransformationen mit je drei Datenmodelltransformationswerkzeugen (siehe Kapitel 3). Die Transformationsfälle zwischen den gewählten Anwendungsfällen und ihren festgelegten Zielmodellen werden von nun ab wie folgt benannt:

- **Transformationsfall I:** Denkmaldaten - INSPIRE *Protected Sites*
- **Transformationsfall II:** Schutzgebiete - INSPIRE *Protected Sites*
- **Transformationsfall III:** AAA 6.0 CityGML 1.0 - INSPIRE *BuildingsCore3DLoD2*

¹GetCapabilities ist ein Dienst zur Übermittlung von Metadaten eines WFS.

²DescribeFeatureType ist ein Dienst zur Beschreibung der Eigenschaften von vorhandenen FeatureTypes eines WFS.

Als Ausgabeformate der ETL-Tools sind GML 3.2.1 (I & II) bzw. BuildingsCore3DLoD2 geplant.

4.2 Analyse der Heterogenität zwischen den Datenmodellen

Einen wichtigen Faktor, mit dem sich eine Aussage über die Schwierigkeit der Datenmodelltransformation im jeweiligen Transformationsfall treffen lässt, stellt das Maß der Ähnlichkeit (Homogenität) zwischen Quell- und Zielmodell dar, oder anders formuliert: Die Heterogenität zwischen den Modellen (siehe Abschnitt 2.3.1). Je heterogener sich zwei Modelle zueinander verhalten, desto komplexer fällt in der Regel auch die Transformation zwischen beiden Modellen aus. Im Gegensatz gilt, je ähnlicher sich zwei Modelle sind, desto einfacher lässt sich eine Transformation zwischen beiden durchführen - in der Theorie.

Die in den nachfolgenden Abschnitten 4.2.1 bis 4.2.3 erläuterten Heterogenitäten werden später zur Analyse der Anwendungsschemata in den Abbildungstabellen herangezogen und tragen auch dazu bei, die Grenzen der Datenmodelltransformation an sich zu ermitteln.

4.2.1 Heterogenität der korrespondierenden Elemente

Die Heterogenität lässt sich im Detail am Besten beschreiben, indem die Analyse direkt die korrespondierenden Elemente aus Quell- und Zielschemata betreffend vorgenommen wird. Es handelt sich also um eine Analyse der semantischen Relationen (siehe Abschnitt 2.4.1). In der Dissertation von Fichtinger (2011, S.101ff.) wurde eine Klassifikation der semantischen Relationen und Heterogenitäten entwickelt und auf einen komplexen Transformationsfall angewandt. Die Klassifikation wurde in einem grafischen Schema dargestellt (Abbildung 4.1) und tabellarisch beschrieben [Fichtinger 2011, S.105ff.]. Im Abschnitt 5.1 wird die Klassifikation auf die Transformationsfälle der vorliegenden Arbeit übertragen, um die Lösungsstrategien der bewerteten ETL-Tools für bestimmte Fälle von Heterogenitäten zu ermitteln.

4.2.2 Heterogenität im Umfang der Modelle

Es liegt auch eine Form der Heterogenität vor, wenn Quell- bzw. Zielmodell von erheblich unterschiedlichem Umfang sind. Hierbei ist ersichtlich, dass nicht beide Modelle den gleichen Umfang an Semantik abdecken können und es somit bei einer Transformation zum Verlust von Information respektive Semantik kommen kann. Gegebenenfalls kann eine Transformation dadurch erheblich erschwert werden, dass für Teile des umfänglicheren Modells keine korrespondierenden Elemente gefunden werden können.

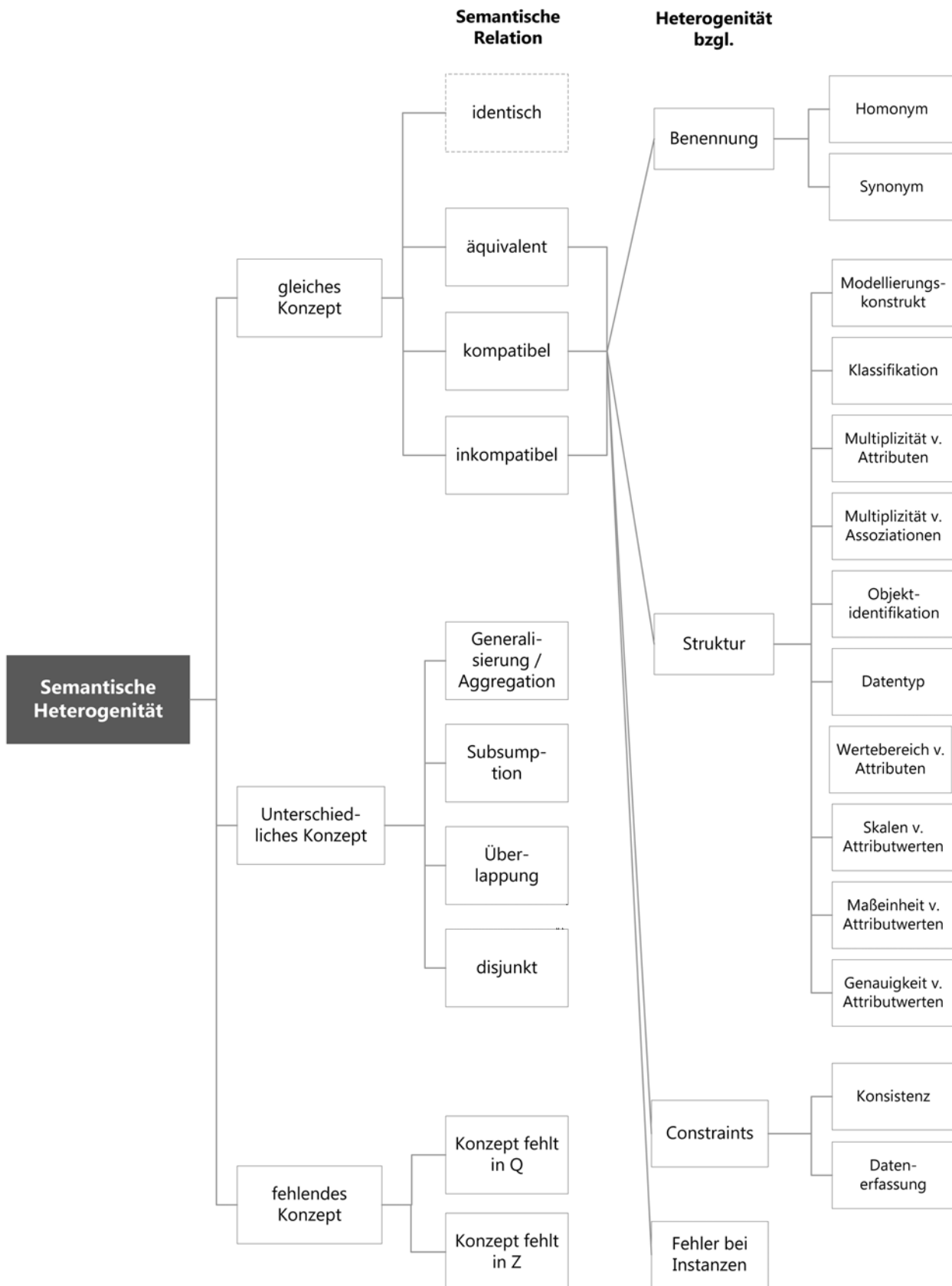


Abb. 4.1 – Klassifikation der semantischen Relationen und Heterogenitäten [Fichtinger 2011, S.104]

4.2.3 Geometrische Heterogenität

Diese Form der Heterogenität zwischen Quell- und Zielmodell spielt für die vorliegenden Transformationsfälle I und II keine große Rolle, da die Geometrie identisch ist und einfach übernommen werden kann. Im Fall 3 ergeben sich jedoch einige Änderungen, die sich über die Heterogenität der korrespondierenden Elemente erfassen lassen (siehe Abschnitt 4.2.1). Darüber hinaus muss in allen drei Fällen eine Transformation des Koordinatenreferenzsystems aus dem in Deutschland amtlichen System heraus erfolgen.

4.3 Gestaltung von Abbildungstabellen

Der Lösungsansatz einer Datenmodelltransformation besteht im Wesentlichen darin, zu ermitteln, welche Elemente aus dem Quellmodell auf welche Elemente des Zielmodells abgebildet werden können, trotz der möglichen Heterogenität zwischen den Modellen. Man bezeichnet in der Geoinformatik das in-Beziehung-setzen zweier Schemata durch semantische Relationen im Kontext der Datenmodelltransformation auch als Schema-Mapping³.

Es bietet sich an, die Relationen und funktionalen Konstrukte nicht nur unmittelbar innerhalb der Softwareumgebung eines ETL-Tools zu definieren, sondern schon vorab extern festzulegen. Dies hat den Vorteil, dass die Abbildung plattformunabhängig stattfindet und somit auch außerhalb einer spezifischen Softwareumgebung wiederverwendet werden kann. Ein weiterer Grund ist, dass diese plattformunabhängigen Abbildungen bereits eine einfache Form der Strukturierung beinhalten und deshalb gegebenenfalls von Software (z.B. Microsoft Excel) eingelesen und weiterverarbeitet werden können. Damit können Abbildungen u.a. auch zwischen Anwendern ausgetauscht werden. Im Rahmen der Forschungsarbeit von Nissen et al. (2011) wird eine Vorgehensweise zur vereinheitlichten Festlegung der Abbildungen in tabellarischer Form vorgestellt. Darin wurde eine abstrakte Vorlage für eine Abbildungstabelle entworfen, welche die zu mappenden Elemente in verschiedenen Spalten einer Zeile gegenüberstellt. Die Vorlage wird in Abbildung 4.2 wiedergegeben.

In der exemplarischen Abbildungstabelle erfolgt eine Abgrenzung der Modelle in einen linken Bereich (graue Spalten) und rechten Bereich (gelbe Spalten). Dazwischen drückt eine Spalte (grün) aus, mit welcher Funktion oder welchem Konstrukt die Abbildung zwischen den Elementen vorgenommen werden soll. Die zugrunde liegenden Objekthierarchien können ebenfalls abgebildet werden, indem Klassen, Attribute und Attributwerttypen in vorgegebener Reihenfolge zeilenweise einander untergeordnet werden. Nach der Kopfzeile (enthält die Beschriftungen zu den Spalten) enthält die Tabelle in der ersten Zeile den (ersten) Objekttypen, dann die zugehörigen Attribute. Treten komplexe Attribute auf, werden sie in die Spalte der Objekttypen eingegliedert und deren Attribute werden

³Das Schema-Mapping bezeichnet den Vorgang zur Ermittlung der semantischen Relationen und Funktionen zur Abbildung zwischen Schemata und ein Mapping-Schema das niedergeschriebene Ergebnis daraus.

| Target Feature class.type | Target Feature attribute | Target Feature documentation | Target Attribute value | Mapping expression | Source Feature class.type | Source Feature attribute | Source Feature documentation | Source Attribute value | Remarks |
|---------------------------|--------------------------|------------------------------|------------------------|--------------------|---------------------------|--------------------------|------------------------------|------------------------|---------|
| <class> <type1> | | | | | <class> <type2> | | | | |
| | <attr2> | | | <func1> | | <attr5> | | | |
| | | | | | | <attr6> | | | |
| | <attr3> | | <value1> | <func2> | | <attr7> | | <value3> | |
| | | | <value2> | <func3> | | <attr8> | | <value4> | |

Abb. 4.2 – Template für Abbildungstabellen zum Schema-Mapping [Nissen et al. 2011, S.3]

wiederum untereinander aufgeschlüsselt. In manchen Anwendungsschemata können komplexe Attribute mehrmals verschachtelt vorkommen (komplexe Attribute enthalten komplexe Attribute). Des Weiteren besteht mit zusätzlichen Spalten beliebig viel Platz für Erläuterungen, Anmerkungen oder Erweiterungen. Die Abbildungstabellen in dieser Form sind auch nicht darauf beschränkt, nur ein Anwendungsschema miteinzubeziehen (siehe Abschnitt 6.1.3).

Zu allen drei Transformationsfällen stehen Abbildungstabellen (syn. Mapping-Schema) schon vorab zur Verfügung. Die Gestaltung von Abbildungstabellen war somit nicht Aufgabe dieser Arbeit. Die Abbildungstabellen für die Transformationsfälle I & II wurden vom LDBV zur Verfügung gestellt. Das Mapping-Schema für den Transformationsfall III entstammt der AdV und wurde im Rahmen der Bachelor's Thesis von Popp (2015) bereits auf seine Tauglichkeit hin überprüft.

Aus Gründen der Übersichtlichkeit sollen nachfolgend nicht die Abbildungstabellen wiedergegeben werden; diese sind in vorliegender Fassung vollständig im digitalen Anhang (Anhang B) enthalten und dort folgendermaßen benannt (Tabelle 4.1):

| Transformationsfall | Benennung der Abbildungstabelle |
|-------------------------|---|
| Transformationsfall I | Abbildung_INSPIRE-PS_Denkmal_2015-02-13 |
| Transformationsfall II | Abbildung_INSPIRE-PS_Schutzgebiete_2015-01-23 |
| Transformationsfall III | MappingTable_AAA-CityGML10_20150522 |

Tab. 4.1 – Benennung der Abbildungstabellen

In den Abschnitten 5.1.1 bis 5.1.3 werden die Abbildungstabellen näher analysiert. Es kann die Klassifikation der semantischen Relation nach Abbildung 4.1 vorgenommen und festgestellt werden, wie viele Elemente an der Bildung einer Relation beteiligt sind.

Die Schwierigkeit wird vor allem dadurch bedingt, die Elemente mit korrekten semantischen Relationen zu identifizieren und geeignete funktionale Konstrukte aus den Funktionsbibliotheken der Tools für die Abbildungen zu wählen, falls Werte nicht im originären Zustand übernommen werden können. In den ausgewählten Tools werden hierfür unterschiedliche Ansätze für Abbildungsmechanismen verfolgt, worauf in der Bewertung der Tools näher eingegangen wird (siehe Abschnitte 6.3.1 bis 6.3.2).

Als Zielsysteme für die Reprojektierung der Koordinatenreferenzsysteme wurden in den Spezifikationen folgende Systeme identifiziert:

Transformationsfälle I & II: 2D geodetic in ETRS89 on GRS80 [INSPIRE DS PS, S.28].

http URI Identifier: <http://www.opengis.net/def/crs/EPSG/0/4258>

Transformationsfall III: 3D geodetic in ETRS89 on GRS80h [INSPIRE DS BU, S.132].

http URI Identifier: <http://www.opengis.net/def/crs/EPSG/0/4937>

In Kapitel 5 werden die Ergebnisse der in der Praxis durchgeführten Datenmodelltransformationen aufbereitet dargestellt, unterschieden nach den verwendeten ETL-Tools und Transformationsfällen. Dabei wird auch auf Besonderheiten und Auffälligkeiten, sowie Fehler eingegangen, die bei der Bewertung in Kapitel 6 relevant sind. Die kompletten erzeugten Datensätze sind dem digitalen Anhang (Anhang B) zu entnehmen.

4.5 Erstellung eines Bewertungsschemas

Im Anschluss an die durchgeführten Datenmodelltransformationen soll ein Bewertungsschema entworfen werden, um die Eignung der ETL-Tools hierfür zu prüfen. Die Beurteilung ergibt sich überwiegend aus den in der vorliegenden Arbeit gewonnenen Erkenntnissen und Erfahrungswerten während der durchgeführten Transformationen. Zusätzlich sollen aber auch Aspekte beachtet werden, die bei den vorliegenden Transformationsfällen keine Rolle spielten. Außerdem soll das Schema nicht nur zur Bewertung der drei vorliegenden Tools angewandt werden können, sondern möglichst auch zur Feststellung der Qualifikation weiterer Datenmodelltransformationswerkzeuge ausgelegt sein.

Die Bewertung stützt sich auf eine Reihe von Bewertungskriterien, die in den Abschnitten 6.2.1 und 6.2.2 festgelegt werden. Zunächst werden die Grundlagen und Ergebnisse aus anderen Forschungsarbeiten angeführt, die bei der Festlegung der Kriterien hinzugezogen wurden.

Speziell für den Kontext von INSPIRE wurden durch Beare et al. (2010) Analysen durchgeführt, um die Eignung von Schema-Beschreibungssprachen (engl. Schema Description Language), Abbildungssprachen und Datenmodelltransformationswerkzeugen zur Durchführung von Transformationen in INSPIRE-konforme Anwendungsschemata zu untersuchen. Hierbei wurden die Untersuchungsgegenstände beschrieben und anhand unterschiedlicher Aspekte klassifiziert sowie bewertet. Legler und Naumann (2007) beschreiben nicht nur die Klassifikation von Transformationsfällen und Korrespondenzen, sondern haben auch Tools zur Definition von Abbildungen getestet und evaluiert.

Die Forschungsarbeit von Kutzner und Eisenhut (2010) hatte einen großen Einfluss auf die vorliegende Arbeit. Auch aus den dort vorgestellten Sachverhalten konnten Aspekte extrahiert werden, die bei Datenmodelltransformationen zu beachten sind.

In der Dissertation von Fichtinger (2011) wurden semantische Transformationen im Kontext von INSPIRE mit dem Tool HALE durchgeführt und bewertet. In vorherigen Abschnitten wurden schon einige im Zuge der Dissertation entwickelte Resultate, vornehmlich Klassifizierungsschemata, angewendet. Es wurden auch „Anforderungen an Lösungen zur semantischen Transformation“ [Fichtinger 2011, S.67ff.] festgestellt und benötigte Transformationsfunktionen ermittelt.

Unger (2012) führte in ihrer Diplomarbeit ebenfalls eine semantische Modelltransformation im Kontext von INSPIRE mit dem Tool HALE durch. Die Software wurde nicht ausführlich bewertet, jedoch finden sich zahlreiche Hinweise, die für die Bewertung im Zuge dieser Arbeit hilfreich sind.

Im Rahmen der Bachelor's Thesis von Popp (2015) erfolgte eine semantische Transformation von CityGML im Kontext von INSPIRE, durchgeführt mit der Software FME. Aus diesem Anlass wurde der Transformationsfall III hier nicht mit FME realisiert. Die gewonnenen Erkenntnisse aus der Thesis unterstützen gleichermaßen die Entwicklung eines Bewertungsschemas.

Das Bewertungsschema wird in zwei äquivalenten Konzepten entworfen. In drei Bewertungskatalogen werden zunächst die Kriterien zur Evaluierung der Ergebnisse und Kriterien zur Bewertung der Tools entwickelt (siehe Abschnitte 6.2.1 und 6.2.2). Diese Kriterien werden dann in Tabellen zusammengefasst und dort unter Anführung von Gründen bewertet. Anschließend erfolgt noch die grafische Strukturierung der Kriterien in Verzweigungsdiagrammen, um die Zusammenhänge der Kriterien durch Visualisierung besser darstellen zu können. Mit beiden Varianten ist eine identische Bewertung möglich.

Kapitel 5

Darstellung der Transformationsergebnisse

Im praktischen Teil der Thesis wurden Datenmodelltransformationen durchgeführt (wie in Kapitel 4 beschrieben). Die Ergebnisse der Transformationen und Beobachtungen bezüglich der ETL-Tools werden in diesem Kapitel wiedergegeben und aufbereitet. Die erzeugten Datensätze und Programmdateien der Tools sind im digitalen Anhang (siehe Anhang B) enthalten.

5.1 Analyse der Abbildungstabellen

Die semantischen Relationen (SR), auf welche die Funktionen angewandt werden sollen, können nach der in Fichtinger (2011, S.101ff.) entwickelten Klassifikation, wie in den folgenden drei Abschnitten dargestellt, unterschieden werden.

5.1.1 Analyse der Abbildungstabelle Denkmalschutzgebiete - Protected Sites

Transformationsfall I

Das Anwendungsschema Denkmal BLfD kennt drei FeatureTypes mit teils unterschiedlichen Attributen, wohingegen im Anwendungsschema Protected Sites nur ein FeatureType geführt wird. Demzufolge müssen für alle originären FeatureTypes separat Abbildungsrelationen gebildet werden. Diese erfolgen gleichförmig, was bedeutet, dass für jeden FeatureType dieselben Zielelemente ausgewählt wurden und folglich, die aus der Transformation erzeugten Objektinstanzen, konsistent sind.

Folgende Attribute des originären Anwendungsschemas werden zur Bildung von semantischen Relationen mit Elementen des Zielschemas herangezogen: `geometry`, `uId`, `adresse` und `gemeindeName` für den FeatureType `Baudenkmaeler`, zusätzlich `name` für `Bauensembles` und `aktenNummer` für `Bodendenkmaeler`. Der Informationsgehalt der restlichen Attribute entfällt im Zielschema. Alle weiteren Belegungen mit Werten (Wertzuweisungen) beruhen auf Angaben aus der Abbildungstabelle, oder sind obligatorisch mit dem `VoidReasonValue` zu belegen, das Zielattribut vom Stereotyp «voidable» ist.

Gemäß der Abbildungstabelle werden zwei Transformationsfunktionen benötigt: Wertzuweisung (WZ)

und Datentypkonvertierung (DT).

- **SR.I.1 (DT): Äquivalentes Konzept im Quellmodell mit unterschiedlichem Datentyp**
Und auch unterschiedlicher Benennung. Dieser Fall betrifft die Übertragung der eindeutigen und persistenten Identifikatorzuweisung aus dem Quellmodell mit einer Datentypkonvertierung von Integer nach String.
- **SR.I.2 (WZ): Äquivalentes Konzept im Quellmodell**
Beispielsweise soll im Zielmodell ein geographischer Name vergeben werden. Die Objekte des FeatureTypes Bauensemble wurden mit einem Eigennamen benannt und der Attributwert kann aus den originären Daten übernommen werden.
- **SR.I.3 (WZ): Unterschiedliches Konzept im Quellmodell mit unterschiedlicher Strukturierung**
Bei den FeatureTypes Baudenkmaeler und Bodendenkmal er erfolgt im originären Modell keine Benennung, sondern eine Adressierung, bestehend aus mehreren Attributen. Durch Zusammenführung der betreffenden Attributwerte kann die Adresse die Benennung mit einem Eigennamen im Zielmodell ersetzen.
- **SR.I.4 (WZ): Fehlendes Konzept im Quellmodell**
Es fehlt ein korrespondierendes Quellattribut. Attribute mit dem Stereotyp «voidable» müssen dennoch nach den vorgesehenen Regelungen berücksichtigt werden, insofern ihre Multiplizität ungleich 0 ist.
- **SR.I.5 (WZ): Fehlendes Konzept im Quellmodell mit möglicher Herleitung**
Es fehlt ein korrespondierendes Quellattribut, jedoch kann der Attributwert hergeleitet werden, beispielsweise aus der Klassifizierung der originären FeatureTypes.
(z.B. siteProtectionClassification="archaeological" im Fall Bodendenkmäler)

Aufgrund eines kompatiblen Konzeptes, wird für die Abbildung der Geometrie keine Funktion benötigt. Es handelt sich um ein kompatibles - und nicht identisches - Konzept, da sich die Geometrie, wegen unterschiedlicher Koordinatenreferenzsysteme, in der Struktur des Wertebereichs unterscheidet. Zur Transformation der Koordinatenreferenz wird eine Funktion benötigt. Dieser Sachverhalt wird in der Abbildungstabelle allerdings nicht erwähnt.

5.1.2 Analyse der Abbildungstabelle Naturschutzgebiete - Protected Sites

Transformationsfall II

Das Anwendungsschema Schutzgebiete LfU beinhaltet sieben FeatureTypes die auf einen Feature-Type ProtectedSite im Zielschema abgebildet werden. Die originären FeatureTypes wurden aufgrund von semantischen Ähnlichkeiten zu drei Gruppen zusammengefasst, wobei nicht alle Attributwerte identisch belegt werden. Darauf ist besonders zu achten. Von den originären Attributen werden folgende zur Abbildung auf das Zielschema herangezogen: geometry, id und name. Die verbleibenden

Attribute geben ihren Inhalt nicht an das Zielschema weiter. Ansonsten werden auch in diesem Fall die weiteren Wertzuweisungen nach Angaben aus der Abbildungstabelle vorgenommen und die Attribute mit Stereotyp «voidable» nach den Regeln für fehlende Werte belegt.

Gemäß der Abbildungstabelle werden zwei Transformationsfunktionen benötigt: Wertzuweisung (WZ) und Datentypkonvertierung (DT).

- **SR.II.1 (DT): Äquivalentes Konzept im Quellmodell**

Beispielsweise soll im Zielmodell ein geographischer Name vergeben werden. Auch die FeatureTypes aus Naturschutzes wurden mit Eigennamen benannt und die Attributwerte können in die Zieldaten übernommen werden. (entspricht SR.I.2)

- **SR.II.2 (WZ): Fehlendes Konzept im Quellmodell**

Es fehlt ein korrespondierendes Quellattribut. Attribute mit dem Stereotyp «voidable» müssen dennoch nach den vorgesehenen Regelungen berücksichtigt werden, insofern ihre Multiplizität ungleich 0 ist. (entspricht SR.I.4)

- **SR.II.3 (WZ): Fehlendes Konzept im Quellmodell mit möglicher Herleitung**

Es fehlt ein korrespondierendes Quellattribut, jedoch kann der Attributwert hergeleitet werden, beispielsweise aus der Klassifizierung der originären FeatureTypes.

(z.B. `designation="nationalPark"` im Fall Nationalpark) (entspricht SR.I.5)

Zur Geometrie sind die gleichen Anmerkungen anzuführen, wie im vorherigen Abschnitt zu Transformationsfall I. Da auch ansonsten keine Abweichung zur Abbildungstabelle von Transformationsfall I vorliegt, werden diese Klassifizierungen nicht separat verwendet.

5.1.3 Analyse der Mapping-Tabelle AAA6.0 CityGML1.0 - INSPIRE BuildingsCore3DLoD2

Transformationsfall III

Die Schemata dieses Anwendungsfalls sind sich relativ zu den Fällen I & II gesehen ähnlicher. Ein Building-Objekt mit zwei BuildingParts wird auf ein ebensolches abgebildet. Der Testdatensatz zum Transformationsfall wurde nach den Regeln des GeoInfoDok 6.0 gestaltet, wohingegen die Mapping-Tabelle erst mit dem GeoInfoDok 7.0 veröffentlicht werden wird, welches sich noch im Entwicklungsprozess befindet. In der Bachelor's Thesis von Popp (2015) wurde die Mapping-Tabelle auf ihre Eignung hin untersucht. Dort wurde festgestellt, dass die Tabelle einige Fehler und falsche Benennungen enthält, so dass ihre Nutzung mit einigem Aufwand verbunden ist [Popp 2015, S.24f.]. Zudem wird die Mapping-Tabelle in Roschlaub (2013) kurz vorgestellt und ihre Struktur erklärt. In der Mapping-Tabelle sind sogar insgesamt vier Anwendungsschemata enthalten. Bei den zusätzlichen Schemata handelt es sich um das Anwendungsschema CityGML 1.0 (aus dem das originäre Schema des Transformationsfalls hervorging) und die CityGML INSPIRE Buildings Core3D ADE (welche

eine Erweiterung von CityGML als Anpassung an INSPIRE darstellt; siehe Abschnitt 3.1.3). Das Zielschema wird in der Tabelle im Profil INSPIRE BuildingsExtended3D mit allen LoD angegeben und ist damit deutlich umfangreicher, als das ausgewählte Zielschema, welches darin enthalten ist.

- **SR.III.1 (WZ): Äquivalentes Konzept im Quellmodell**

Zum Zielelement ist ein äquivalentes Element im Quellmodell enthalten, dass sich beispielsweise lediglich durch eine variierende Bezeichnung unterscheidet. Ein Beispiel für einen solchen Fall ist das Quellattribut `gml:id`, das in das komplexe Attribut `inspireID` mit dem Zielattribut `localId` eingeht. (entspricht SR.I.2)

- **SR.III.2 (WZ): Identisches Konzept im Quellmodell**

Der Attributwert des Quellmodells kann in das Zielmodell übernommen werden. Beispielsweise wird die Geometrie der `TerrainIntersection` identisch als `GM_MultiCurve` übernommen.

- **SR.III.3 (WZ): Fehlendes Konzept im Quellmodell**

Es fehlt ein korrespondierendes Quellattribut (z.B. `conditionOfConstruction`). Attribute mit dem Stereotyp «voidable» müssen dennoch nach den vorgesehenen Regelungen berücksichtigt werden, insofern ihre Multiplizität ungleich 0 ist. (entspricht SR.I.4)

- **SR.III.4 (WZ): Fehlendes Konzept im Quellmodell mit möglicher Herleitung**

Es fehlt ein korrespondierendes Quellattribut, jedoch ist der Zielattributwert aus anderen Quellen bekannt (z.B. `inspireId.Identifier.namespace`). (entspricht SR.I.5)

- **SR.III.5 (WZ): Kompatibles Konzept im Quellschema mit unterschiedlichen Modellierungskonstrukten** Die Geometrie wird an die INSPIRE-Kodierung angepasst:

Die `CompositeSurfaces` werden nicht mehr nach den Gebäudeflächen angegeben, sondern als `gml:posList` einer `GM_MultiSurface` im LoD 2.

Zudem wird das Koordinatenreferenzsystem in eines der von INSPIRE vorgesehenen Systeme reprojiziert [INSPIRE BU 2013, S.132].

5.2 Ergebnisse für Feature Manipulation Engine

5.2.1 Transformationsansatz von FME

Die Abbildungen werden auf Ebene der Anwendungsschemata definiert und somit handelt es sich um horizontale Datenmodelltransformationen. Dazu werden mit zwei Funktionen (Reader, Writer) die Schemata eingeladen. Anschließend können die geladenen `FeatureTypes` der Schemata und deren Attribute betrachtet werden. Die Transformationsrichtung innerhalb eines Workspaces wird immer unidirektional angelegt. Da in Reader und Writer auch WFS gelesen werden können, ist auch eine online durchgeführte Transformation möglich. Jedoch handelt es sich bei FME um eine Desktop-Applikation und keinen Webdienst, so dass die Transformation streng genommen nicht on-the-fly, durch Abruf

des Nutzers über eine Webschnittstelle, möglich ist. Nach Fertigstellung der Abbildungsdefinition wird ein Quelldatensatz im Reader eingeladen, transformiert und durch den Writer im Zielschema geschrieben.

5.2.2 Transformation der Geometrie mit FME

FME kann die Geometrie der Formate GML 3.1.1, GML 3.2.1, CityGML (alle Versionen) sowie INSPIRE-spezifischer Formate sowohl lesen, als auch bearbeiten. Aus den 459 Funktionen (Transformer) von FME 2015 gehören allein 30 der Rubrik geometrischer Operatoren an, wobei Filterfunktionen und Funktionen für Berechnungen etc. nicht mit hinzugezählt wurden.

Transformationsfälle I & II

Die Geometrie wird vom Reader eingelesen und als *xml:geometry* gespeichert. Zusätzlich werden noch die Attribute *gml:boundedBy*, *gml:location* und *gml:priorityLocation* im gleichen Datentyp hinterlegt. Weiter ist die Geometrie jedoch nicht einsehbar. Das Attribut *gml:boundedBy* wird auch wieder im Resultat geschrieben.

Die einzigst benötigte Transformation war die Reprojektierung des Koordinatenreferenzsystems mit dem FME-Transformer *Reprojector* (Rubrik: Coordinate Systems). Abbildung 5.1 zeigt einen die Geometrie betreffenden Ausschnitt aus einem erzeugten GML-Instanzdokument. Das Koordinatenreferenzsystem wird standardmäßig durch seine EPSG-Nummer angegeben (*srsName*="EPSG:4258"). Die angegebenen *gml:ids* wurden von den originären Features identisch übernommen. In machen Fällen (insbesondere bei Transformationsfall II) kann ein Feature mehrere Geometrien (*gml:Surface*) innerhalb des Knotens *gml:MultiSurface* besitzen. Unterschieden werden diese in der Kennung der *gml:id*, an die eine ganzzahlige Zählung angehängt wird. Im Workspace wurde die Geometrie mit dem Transformer *GeometryValidator* überprüft, mit dem sich zwölf Parameter zur Validierung einstellen lassen (z.B. ob NaN- oder Null-Werte enthalten sind). Dabei wurden von FME keine Fehler festgestellt und auch eine nachträgliche Überprüfung der Resultate mit QGIS¹ und ArcGIS² konnte die Korrektheit der Geometrie bestätigen.

Transformationsfall III *entfällt* (siehe Abschnitt 4.1)

5.2.3 Transformation der Semantik mit FME

Zum Erstellen von Inhalten stehen erheblich mehr Funktionen zur Verfügung, als in der Rubrik Geometrie. In den angewandten Transformationsfällen I & II musste eine Konvertierung des Datentyps von Integer auf String durchgeführt werden. Die Datenkonvertierung nimmt FME automatisch vor,

¹QGIS ist eine weit verbreitete Open-Source-Geoinformationssoftware (verwendete Version: Wien).

²ArcGIS ist eine weit verbreitete proprietäre Geoinformationssoftware (verwendete Version: 10.2.2).

```

- <ps:ProtectedSite gml:id="BLFD_BAUDENKMAELER_2745">
- <ps:geometry>
- <gml:MultiSurface gml:id="BLFD_BAUDENKMAELER_2745_BLFD_GEOMETRY" srsName="EPSG:4258" srsDimension="2">
- <gml:surfaceMember>
- <gml:Surface gml:id="GEOMETRY_e77b3375-bed1-4d09-a379-3185d40a22b2">
- <gml:patches>
- <gml:PolygonPatch>
- <gml:exterior>
- <gml:LinearRing>
- <gml:posList>
48.607862725764 12.8259999153192 48.6079590280001 12.8259387014129 48.6079304073204 12.8258479892684 48.60
12.8259999153192
</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:PolygonPatch>
</gml:patches>
</gml:Surface>
</gml:surfaceMember>
</gml:MultiSurface>
</ps:geometry>

```

Abb. 5.1 – FME: Beispiel für die Ausgabe der Geometrie in den Transformationsfällen I & II

wenn der entsprechende Transformer gewählt wird (StringConcatenator). Strings sind in der Regel auch der Default-Attributtyp in FME (nicht für Geometrie).

Die Reihenfolge der transformierten Features im Instanzdokument entspricht der Reihenfolge der Features im originären Zustand, sortiert nach `gml:id`.

Transformationsfall I

Zur Durchführung der Transformationen genügten fünf verschiedene Typen von Transformern. Diese wurden im Wesentlichen dazu verwendet, neue Attribute anzulegen, Attributwerte aus Quellelementen auf Zielattribute zu übertragen, Filterungen durchzuführen, Attributwerte (String) aus Quellelementen aufzuspalten und neu zusammen zu setzen. Praktischerweise können neu erzeugte Attribute durch den selben Transformer auch gleich mit Werten belegt werden.

Die Elemente des Attributs `legalFoundationDocument` des komplexen Attributtyps «CI_Citation» wurden mit einem *XMLTemplater* erzeugt, da darin enthaltenen Attribute in FME intern als Typ «xml_xml» hinterlegt werden. In dieser Funktion wurden die Attribute als XML-Ausdruck angelegt. Da im Namespace der originären Daten der Character & zweifach enthalten ist, muss dieser XML-konform durch den Ausdruck *&* ersetzt werden. Außerdem muss der Namespace `namespace=,http://www.isotec211.org/2005/gmd` in der Funktion deklariert werden, damit in den Zieldaten der Namespace-Präfix zu den Attributen angegeben wird.

Beim Anlegen eines Writers im Workspace wird durch einen Einstellungsdialog abgefragt, welches Format geschrieben werden soll (außerdem kann die `gml:id` der FeatureCollection an dieser Stelle festgelegt werden). Der Anwender kann/muss aus einer Liste vorgegebener Formate auswählen (z.B. INSPIRE GML). Unter den Parametern des Dialogs kann das zugehörige Anwendungsschema ausgewählt werden, entweder aus den lokal in FME hinterlegten Schemata, oder aus anderer Quelle. Für die Lösung des Problems ist die Bezugsquelle der Auswahl jedoch irrelevant.

Transformationsfall II

Die sieben FeatureTypes werden in der entsprechenden Abbildungstabelle (siehe Abschnitt 4.3) zu drei Gruppen sortiert, da einige der FeatureTypes vereinzelt identische Attributwerte zugewiesen bekommen. FME bietet verschiedene Möglichkeiten an, die Attributbelegung gezielt zu steuern und somit alle FeatureTypes innerhalb eines Workflows bearbeiten zu können. Beispielsweise ist u.a. in den Transformern *AttributeCreator* eine konditionierte Belegung von Attributwerten durch Angabe von Bedingungen möglich (z.B. Unterscheidung der FeatureTypes anhand eines anderen, klassifizierenden Attributwertes). Alternativ kann der Workflow aber auch durch Verzweigungen (z.B. Transformer *AttributeFilter*) aufgespalten werden, um die FeatureTypes mit individuell zugewiesenen Transformern zu bearbeiten. Im schlichtesten (aber möglicherweise unübersichtlichsten) Fall durchläuft jeder FeatureType einen eigenen Workflow, die dann erst im Writer zusammengeführt werden. Das Attribut `legalFoundationDocument` wird in diesem Transformationsfall nicht belegt.

5.2.4 Lösungsstrategien für die klassifizierten semantischen Relationen

In den Abschnitten 5.1.1 und 5.1.2 wurden die semantischen Relationen der angewandten Transformationsfälle klassifiziert. Nachfolgend werden die gewählten Vorgehensweisen zur Abbildung der semantischen Relationen in FME knapp beschrieben und die verwendeten Transformer angegeben:

- **SR.I.1 (DT): Äquivalentes Konzept im Quellmodell mit unterschiedlichem Datentyp**

Die Konvertierung von Integer nach String übernimmt FME automatisch, es muss nur der Wert übergeben werden. Ansonsten stehen eine Reihe von Funktionen zur Verfügung, um Attributwerte zu erzeugen oder herzuleiten.

AttributeCreator

- **SR.I.2 (WZ): Äquivalentes Konzept im Quellmodell**

Schlichte Übertragung des Attributwertes.

AttributeCreator

- **SR.I.3 (WZ): Unterschiedliches Konzept im Quellmodell mit unterschiedlicher Strukturierung**

Alle eingelesenen Attribute können bei der Erzeugung eines Wertes mitwirken. Zudem lassen sich die Attributwerte verknüpfen und manipulieren, alles bequem in einem Editor, der in den Transformern integriert wurde. Bei komplexeren Operationen müssen eingehende Attributwerte möglicherweise erst aufgespalten werden, wie in Transformationsfall I.

StringConcatenator, TestFilter, ListConcatenator, AttributeSplitter

- **SR.I.4 (WZ): Fehlendes Konzept im Quellmodell**

Ist das Zielattribut vom Stereotyp «voidable», wird es einfach erzeugt und direkt gemäß der Regeln von INSPIRE, für fehlende Attributwerte zu Attributen des Stereotyps, die entsprechen-

den Angaben eingefügt.

AttributeCreator

- **SR.I.5 (WZ): Fehlendes Konzept im Quellmodell mit möglicher Herleitung**

Das Zielattribut wird einfach erzeugt und direkt mit einem Wert belegt.

AttributeCreator, XMLTemplater

Zur Reprojektierung der Geometrie wurde ein *Reprojector* verwendet, zur Validierung ein *GeometryValidator* verwendet. Mit einem *SummaryReporter* wurden zudem Berichte über die Transformationsvorgänge gespeichert.

5.2.5 Paradigmenwechsel während der Transformation mit FME

FME ist intern relational strukturiert. Dies bedeutet in erster Linie, dass die eingelesenen Datensätze, obgleich als objektorientiertes Transferformat importiert, in der Software relational gespeichert und in diesem Paradigma auch weiterverarbeitet werden (vgl. auch [Kutzner und Eisenhut 2010, S.68]). Anschließend ist FME jedoch in der Lage, die relationale Strukturierung wieder objektorientiert in eben diesen Transferformaten auszugeben.

Eine wichtige Frage hierbei ist, ob die objektorientierten Prinzipien wie Vererbung bei den Paradigmenwechseln korrekt übertragen werden.

Der Vorgang verhält sich wie nachfolgend beschrieben und in Abbildung 5.2 (links) dargestellt. Zunächst werden die eingelesenen Daten als sogenannte formatierte Attribute gespeichert und in generische FME Attribute transformiert. Dann wird aus den FME Attributen ein zweiter Datensatz formatierter Attribute im Zielschema erzeugt (z.B. «xml_date» entspricht «fme_buffer»). Das bedeutet, dass ein Mapping nicht direkt zwischen den Elementen aus Quell- und Zielschema generiert wird, sondern FME sein eigenes Format zwischenschaltet. Damit führt FME nicht eine Transformation aus, sondern gleich zwei: Eine Transformation vom Quellschema in das FME interne Schema, und von diesem eine Transformation in das Zielschema. Ein Spezialfall ist bei der Verarbeitung von Geometrien gegeben, bei dem FME zwei generische Attribute erzeugt: «fme_type» und «fme_geometry». Ersteres spezifiziert die Geometrien und zweiteres enthält die Koordinaten. Multiple Geometrien eines einzelnen Features werden dabei als aggregierte Geometrien aufgefasst. In Abbildung 5.2 (rechts) ist das FME Geometriemodell zu sehen.

Das Datenmanagement eines Workspaces zur Laufzeit erfolgt durch den FME Feature Store (FFS). Während eine Transformation ausgeführt wird, legt FME in einem temporären Ordner³ diverse Dateien an. Die meisten dieser Dateien (FFS-Dateien) sind leer, oder möglicherweise ist der Inhalt auch vor dem Anwender verborgen. Am relevantesten erscheinen zwei erzeugte Dokumente, die nach Beendigung der Transformation umgehend wieder von FME gelöscht werden. Eines von beiden Dokumenten ist ein XFMLs⁴-Dokument, in welchem FME die eingelesenen Elementstrukturen in das

³Der Dateipfad in Windows 7 lautet: Dokumente>AppData>Local>Temp.

⁴XFMLs ist ein Skript-Format zur Auswertung und Verarbeitung von XML-Dokumenten.

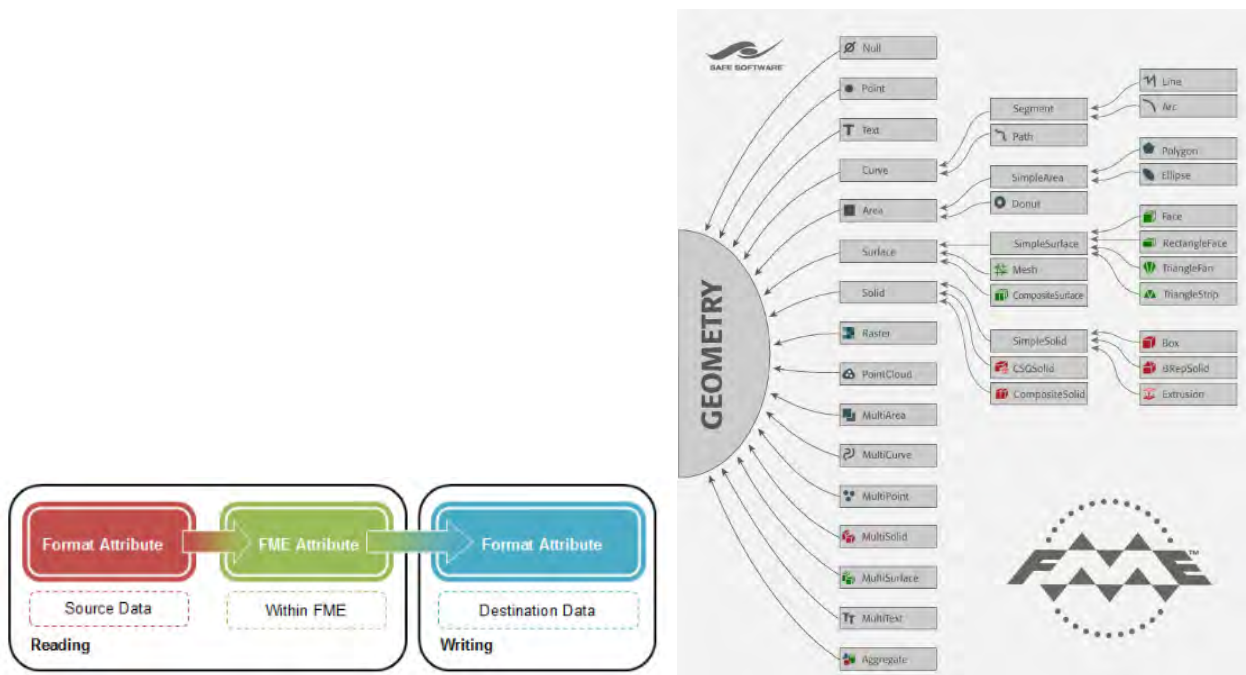


Abb. 5.2 – FME: Interne Handhabung von Attributen (links) und FME Geometriemodell (rechts)
(c) Safe Software Inc.

relationale FME Feature Model dergestalt abbildet (siehe Abbildung 5.3), dass dadurch die eingelesenen objektorientierten Strukturen interpretiert werden. Somit kann FME die Reihenfolge der Knoten festlegen und Geometrien sowie Inhalte zuordnen. Das Dokument gibt zudem das Mapping zwischen den eingelesenen Schemata und dem internen Schema wieder.

```
<?xml version="1.0" encoding="UTF-8"?>
<xfMap xmlns="http://www.safe.com/xml/xfMap" xmlns:gts="http://www.isotc211.org/2005/gts" xmlns:blfd="http://www.denkmal.bayern.de" xmlns:gmd="http://www.isotc211.org/2005/gmd">
  <reference-map>
  </reference-map>
  <reference-content-map>
  </reference-content-map>
  <feature-map multi-feature-construction="true">
  </feature-map>
  <feature-content-map>
  </feature-content-map>
</xfMap>
```

Abb. 5.3 – FME: Zwischenspeicherung von Elementstrukturen mit XFMaps

Das zweite Dokument ist ebenfalls ein XML-Dokument, dass den gesamten Workspace wiedergibt, jedoch auskommentiert wurde. Im ersten Abschnitt werden darin Kennwerte zum Workspace gespeichert (z.B. die Pfade zu den eingelesenen Datensätzen). Der zweite Abschnitt beinhaltet die Datensätze und von FME angelegte Strukturen.

Dabei handelt es sich u.a. um <DATA_TYPES> (u.a. Reader, Writer), <METAFILE_PARAMETERS> (z.B. um das Ausgabeformat festzulegen), <GEOM_TYPES> (überträgt auf das FME Geometriemodell), <FEATURE_TYPES> (bilden die eingelesenen FeatureTypes FME intern ab) und <FEAT_ATTRIBUTES> (beinhaltet die Attribute der FeatureTypes).

Das Mapping (<TRANSFORMER>, <XFORM_ATTR>) wird einzeln für jeden Transformer in <OUTPUT_FEATS> gespeichert. Einige Beispiele sind Abbildung 5.4 dargestellt. Jeder FeatureType und Transformer bekommt einen eindeutigen Identifikator zugewiesen (IDENTIFIER=""), der mit einem



Abb. 5.4 – FME: Elemente zur relationalen Strukturierung

Integer-Wert belegt wird. Durch <FEAT_LINKS>, welche ihrerseits wiederum einen Identifikator besitzen, wird die Abfolge der Funktionen im Workspace festgelegt, indem <SOURCE_NODE> (Quelle) und <TARGET_NODE> (Ziel) identifiziert werden. Dadurch kann die Reihenfolge der Prozessierung identifiziert werden. Das Beispiel für einen <FEAT_LINK> aus Abbildung 5.4 verweist auf die Quelle mit dem Identifikator "27" und das Ziel mit dem Identifikator "39". Beides sind Transformatoren, wobei der Quelltransformator vor dem Zieltransformator abgehandelt wird und beide im Workspace unmittelbar nacheinander geschaltet wurden. Jedem Transformer ist ein <OUTPUT_FEAT> nachgestellt, dass alle durchlaufende Attribute anzeigt. Im letzten Abschnitt des Dokuments hinterlegt FME Makros.

```

<FEAT_ATTRIBUTE ATTR_NAME="legalFoundationDocument.CI_Citation.title" ATTR_TYPE="xml_buffer" ATTR_HAS_PORT="true" ATTR_VALUE=""/>
<FEAT_ATTRIBUTE ATTR_NAME="legalFoundationDocument.CI_Citation.alternateTitle{}" ATTR_TYPE="xml_buffer" ATTR_HAS_PORT="true" ATTR_VALUE=""/>

```

Abb. 5.5 – FME: Repräsentation der Attribute zur Laufzeit

Nach ausgiebiger Analyse der Resultate ist festzustellen, dass FME die Objektorientierung in den Zielformaten korrekt ausgibt (siehe digitaler Anhang B).

5.2.6 Performance von FME

FME führt jedes eingelesene Feature einzeln für sich durch die Prozessierung, so dass keine Wechselwirkungen zwischen den Features entstehen. Dies hat Vorteile in der Performance, da nicht alle Features des Datensatzes während der Laufzeit im Arbeitsspeicher gehalten werden müssen. Es ist aber auch möglich FME anzuweisen, mehrere Features gemeinsam in einer Gruppe zu transformie-

ren, dies ist hier jedoch nicht geschehen.

| Anz. Features | Zeit [sec.] | Spitzenlast [KB] | Durchschnittslast [KB] | Dateigröße [KB] |
|--|-------------|------------------|------------------------|-----------------|
| Verwendete Maschine: Intel(R) Core(TM) i5-2430M CPU 2.40 GHZ 4,00 GB RAM (64 Bit) | | | | |
| Transformationsfall I: 3 FeatureTypes | | | | |
| 10 | 1,2 | 151.640 | 128.432 | 21 |
| 50 | 1,6 | 151.044 | 128.044 | 105 |
| 143.219 | 1390,7 | 2.574.764 | 517.176 | 299.549 |
| 10 | 1,2 | 155.512 | 130.764 | 82 |
| 50 | 1,5 | 155.524 | 130.784 | 248 |
| 870 | 5,5 | 160.604 | 135.276 | 4.660 |
| 10 | 1,2 | 154.220 | 131.316 | 27 |
| 50 | 1,3 | 154.300 | 131.700 | 127 |
| 49106 | 178,8 | 1.165.880 | 1.141.696 | 136.722 |
| Transformationsfall II: 7 FeatureTypes | | | | |
| 10 | 3,2 | 153.804 | 129.180 | 1.207 |
| 252 | 5,2 | 165.916 | 139.916 | 6.405 |
| 10 | 1,4 | 154.800 | 131.788 | 333 |
| 10 | 1,1 | 153.616 | 131.648 | 56 |
| 2 | 1,3 | 155.476 | 132.060 | 348 |
| 5 | 1,6 | 156.868 | 131.680 | 895 |
| 18 | 2,9 | 162.800 | 136.540 | 3.535 |
| 10 | 1,2 | 150.952 | 128.840 | 60 |
| 10 | 1,3 | 151.968 | 129.344 | 381 |

Tab. 5.1 – FME: Kennwerte zur Performance

In Tabelle 5.1 werden Kennzahlen zur Performance von FME ermittelt. Dazu wurden mehrere Transformationen mit verschiedenen großen Datensätzen durchgeführt, zu beiden Transformationsfällen und für jeden FeatureType separat. Die Spalten der Tabelle treffen von links nach rechts folgende ermittelte Angaben: Anzahl der Features der Datensätze, benötigte Zeit zur Transformation, Spitzenlast des Arbeitsspeichers während der Prozessierung, durchschnittliche Last des Arbeitsspeichers während der Prozessierung und Dateigröße der Datensätze. Die Werte in den Spalten zwei bis vier sind Mittelwerte, gebildet aus Einzelwerten mehrerer Durchläufe (mind. 3). Die Anzahl der Features wurde variiert in Kategorien geordnet. Diese setzen sich folgendermaßen zusammen (in Klammern ist die Anzahl der Features angegeben):

(Transformationsfall I) Baudenkmal (10,50,143219), Bauensembles (10,50,870) und Bodendenkmäler (10,50,46106).

(Transformationsfall II) Biosphärenreservate (10,252), Fauna-Flora-Habitat-Gebiete (10), Landschaftsschutzgebiete (10), Nationalparks (2), Naturparks (5,18), Naturschutzgebiete (10) und Vogelschutzgebiete (10).

Gemittelt ergibt sich ein Leistungsvermögen für Transformationsfall I von 0,05 Sekunden Laufzeit je verarbeitetem Feature und 0,02 Sekunden Laufzeit je KB⁵ der Dateigröße des verarbeiteten Datensatzes. In Fall II zeigte sich eine Laufzeit von 0,22 Sekunden je Feature und 0,01 Sekunden je KB Dateigröße. Als Faktor, der die Laufzeit der Software maßgeblich beeinflusst, kann der Umfang der Geometrie eines Features ausgemacht werden. Dieser zeichnet sich auch überwiegend für die Dateigröße des Datensatzes verantwortlich. Die Grundlast im Arbeitsspeicher der *workbench.exe* beträgt weniger als 340.000 KB. Während Ausführung der Transformation werden die Daten erst eingeladen und auf das relationale interne Konzept abgebildet. Im Zuge dessen wird ein zweiter Prozess *fme.exe* ausgeführt, der weitere Lasten erzeugt. Bei etwa 6000 eingelesenen Features überschreitet die Last im Arbeitsspeicher die Last der *workbench.exe* und erreichte bei etwa 40.000 die 100.000 KB.

5.2.7 Berichtswesen und Validierung in FME

FME erzeugt während der Durchführung von Transformationen einen ausführlichen Report, der anschließend als Textdokument im selben Verzeichnis wie die resultierende Datei vorliegt. Aus dem Report können die durchgeführten Arbeitsschritte, etwaige Fehlermeldungen und Zeitstempel sowie Statistiken entnommen werden. Die Validierung der Resultate kann mit der Funktion *XMLValidator* gegen ein Anwendungsschema vorgenommen werden.

Treten während einer Transformation schwerwiegende Fehler auf, so wird diese abgebrochen. Falls das Zieldokument erzeugt werden konnte sind lediglich Warnungen im Report hinterlegt.

5.3 Ergebnisse für GeoKettle

5.3.1 Transformationsansatz von GeoKettle

Die Quelldaten werden mit Input-Funktionen eingelesen. Anwendungsschemata können nicht zur Definition der Abbildungsregeln verwendet werden, da hierfür keine Funktion implementiert wurde. Somit kann die Transformation ausschließlich auf Ebene der Datenformate und undirektional stattfinden. Da GeoKettle eine reine Desktop-Applikation ist, finden Transformationen auch ausschließlich offline statt. Ähnlich wie in FME werden auch in GeoKettle die Funktionen grafisch repräsentiert und auf der Arbeitsfläche der GUI angeordnet. Die Abfolge der Funktionsausführung wird dadurch geschaltet, dass die Funktionen untereinander verbunden werden. Die Transformation erfolgt also entlang der Verbindungslinien. Die Anordnung mündet in einer oder mehreren Output-Funktionen, welche dann die Zieldaten schreiben.

Nach Einlesen der Daten speichert GeoKettle den Datensatz in einer Tabelle und löst somit eine

⁵Die Einheit ist KiloByte [KB].

womöglich vorhandene Objektorientierung vollständig auf. Jedes Feature wird dabei innerhalb einer Zeile geschrieben und die zugehörigen Attributwerte in den Spalten der Zeile hinterlegt. Die Bezeichnung des FeatureTypes selbst geht dabei vollständig verloren.

Möglicherweise können die Abbildungsregeln extern definiert und in GeoKettle hinzugeladen werden (z.B. im Format CSV). Jedoch soll das ETL-Tool selbst auf seine Fähigkeiten zur Definition und Durchführung von Datenmodelltransformationen geprüft werden, weshalb darauf nicht näher eingegangen wird.

5.3.2 Import von Daten

Zum Input von Daten stehen insgesamt 30 Funktionen bereit. Zum Einlesen der vorliegenden Quelldaten kommen zwei Funktionen in Frage. Die Funktion *GML File Input* kann nur Daten der GML-Version 3.1.1 lesen. Das Einlesen aller Datensätze des LfU und des CityGML-Testdatensatzes resultierte in mehreren Fehlermeldungen, teilweise abgebildet in Abbildung 5.6.

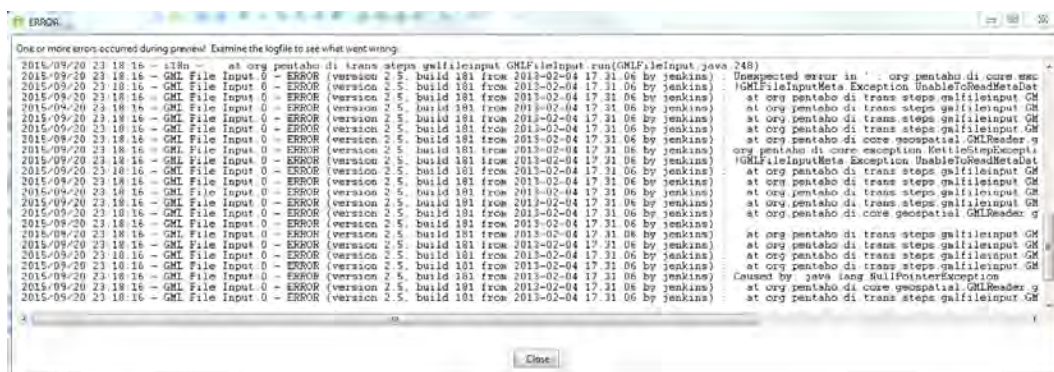


Abb. 5.6 – GeoKettle: Fehlermeldung beim Einlesen der Daten des LfU

Die Datensätze des BLfD konnten hingegen in GML 3.1.1 eingelesen werden. Die Funktion *OGR Input* stellt die zweite Möglichkeit zum Einlesen der Datensätze dar. OGR ist eine Open-Source-Bibliothek von Simple Feature Types der Programmiersprache C++. Mit dieser Funktion konnten alle Datensätze gelesen werden, auch in GML 3.2.1. Eine weitere Funktion, *Get data from XML*, benötigt einen Input in Form von XPath und wird deshalb ausgeschlossen.

In der verwendeten Version der Software besteht außerdem die Möglichkeit, Daten mit der Funktion *WFS Input* über einen WFS-Service zu beziehen. Dazu muss lediglich die URL der GetCapabilities-Request angegeben werden. Der Aufruf des WFS-Services des LfU resultierte jedoch in einer umfangreichen Fehlermeldung (siehe Abbildung 5.7).

5.3.3 Transformation der Geometrie mit GeoKettle

GeoKettle kann die Koordinatenreferenz eines Datensatzes nicht selbstständig identifizieren. Hierfür steht die Funktion *Set SRS* bereit, in welcher der Anwender das System manuell eingeben muss.

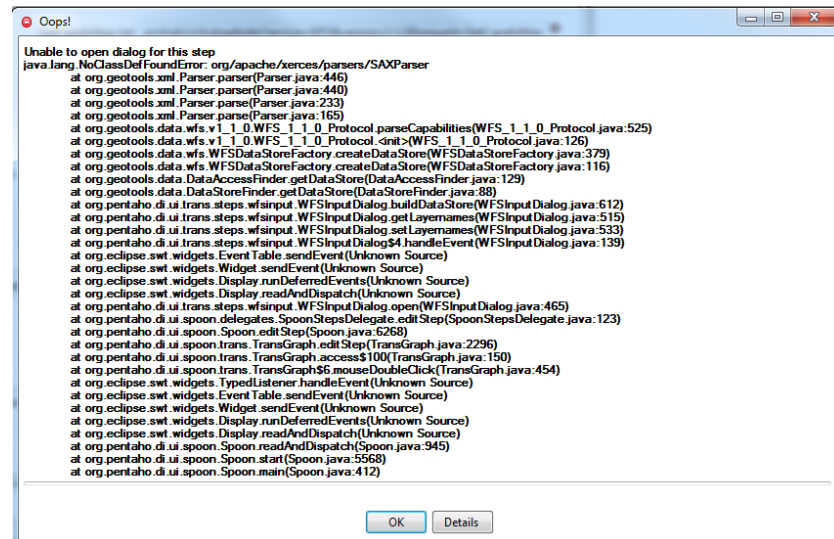


Abb. 5.7 – GeoKettle: Fehlermeldung bei der Abfrage eines WFS

Anschließend wird das System auch in der weiteren Prozessierung automatisch erkannt (z.B. als Input-Parameter für weitere Funktionen). Die Reprojektierung der Koordinatenreferenzsysteme, mit identischen Parametern wie sie auch mit FME und HALE verwendet wurden, lieferte völlig unterschiedliche Ergebnisse. Da die Resultate hierfür aus FME und HALE bis auf kleine Berechnungs- oder Rundungsfehler gleich und korrekt sind, können die Ergebnisse aus GeoKettle eindeutig als falsch belegt werden.

Die Geometrie wird als Koordinatenliste unter vorangestellter Angabe des Geometrietyps (z.B. MultiPolygon) innerhalb einer Zelle der Tabelle gespeichert. Zur Bearbeitung der Geometrie sind in der Funktionsbibliothek keine Funktionen hinterlegt (mit Ausnahme von *Set SRS* und *SRS Transformation*). Folglich kann die Geometrie auch nicht weiter bearbeitet werden.

Transformationsfälle I / II

Da die Geometrie in diesen beiden Fällen keiner weiteren Bearbeitung bedarf, kann sie unverändert in die Zieldaten übernommen werden.

Transformationsfall III

Es wurde jede erdenkliche Funktion von GeoKettle zum Import von CityGML-Daten getestet. In keinem Fall konnte die Geometrie gelesen werden. Mit der Funktion *OGR-Import* wurde der Datensatz zwar eingelesen, als Geometrie wurde dabei lediglich eine Punktkoordinate (GM_MultiPoint) erkannt. Damit fehlen alle relevanten Geometrien des Datensatzes zur weiteren Verwendung. Die Punktkoordinate ist für die Transformation nicht relevant.

5.3.4 Transformation der Semantik mit GeoKettle

Die Attributwerte können nur dahingehend bearbeitet werden, dass die in den Tabellen gespeicherten Attributwerte jedes einzelnen Features manipuliert werden müssen. Zudem muss auch die Struktur der Tabelle komplett überarbeitet werden, indem neue Spalten hinzugefügt und nicht mehr benötigte Spalten gelöscht werden. Mit der Funktion *XML Input* können neue Attribute erzeugt werden - in der Theorie, denn das Anlegen eines einzelnen Attributs in einer neuen Spalte hatte eine Fehlermeldung zur Folge, obgleich in der Voransicht der Funktion das Ergebnis korrekt war. Das Klonen ist nur für Zeilen mit der Funktion *clone row* möglich (mit dem Hintergedanken die darin enthaltenen Attributwerte dann zu überschreiben). Die Funktion *Calculator* ermöglicht das Erzeugen neuer Spalten, wobei die Zugehörigkeit des Attributs zu einem Namespace vorerst nur als Teil des Attributnamens erfolgen kann (z.B. ps:siteProtectionClassification). In den originären Datensätzen der Baudenkmäler ist im Attribut *adresse* in manchen Fällen eine Mehrfachnennung getrennt durch Semikolon enthalten. Die Adressen müssten nach den Vorgaben des BLdD zerlegt und neu zusammengesetzt werden. Nach Anwendung der Funktion *Split fields to rows*, unter Angabe von Semikolon (;) als Trennzeichen, erfolgte jedoch keine Trennung, sondern es wurden drei neue Features (bedingt durch drei Semikolons) mitsamt aller Attribute kopiert. Komplexere Strukturen von INSPIRE (z.B. voidValueReason, inspireID, geographicalName) können nicht erzeugt werden, da die Software keine Möglichkeiten dazu anbietet (vgl. [Vautard 2013, S.35]).

5.3.5 Ergebnis der Transformationen und Berichtswesen

Die Resultate waren in keinem Fall valide zum Zielmodell. Die erzeugten Datensätze sind sogar unbrauchbar. Nach Gebrauch der Funktion *GML File Output* zum Schreiben der Zieldaten wurden folgende Inhalte entfernt: Die *gml:id* wurde an jeder Stelle durch Angabe des Präfix „feature“ mit angehängter ganzzahliger und fortlaufender Nummerierung ersetzt; die *gml:id* ist nun ein „type“-Element und außer dem Namespace „http://www.opengis.net/gml/3.1“ wurden alle weiteren Namespaces nicht berücksichtigt, da sie schlichtweg nicht hinzugezogen werden können und auch nicht intern im Programm hinterlegt sind. Folglich sind auch die Klassen und Attribute keinem Namespace mehr zuweisbar. Die stichprobenartige Untersuchung der Resultate zeigte, dass die Kodierung im Rahmen der möglichen verwendbaren Formate in GeoKettle korrekt ist. Die Funktion *OGR Output* geht gleichermaßen vor, nur dass nun ein Namespace der OGR hinterlegt ist. Dieser Namespace wird auch zwingend verwendet, was die Funktion zur Anwendung für INSPIRE unbrauchbar macht. Zudem unterscheidet sich die verwendete Kodierung von GML 3.1.1 in einigen Punkten (z.B. ist das *gml:MultiPolygon* ein *ogr:geometryProperty*) [OGR GML 2015]. GeoKettle erzeugt zur Laufzeit einen Report. Die darin enthaltenen Informationen sind jedoch nur für Programmierer mit ausreichender Fachkenntnis in Java in Ansätzen brauchbar (siehe Abbildung 5.7).

Da bei der Entwicklung der Workflows zahlreiche Fehler auftraten und viele Funktionen - obwohl benötigt - scheinbar unbrauchbar sind (teilweise wohl im Zusammenhang mit GML 3.1.1), oder ganz

fehlen, wurden die Ansätze zur Verwirklichung von Datenmodelltransformationen in GeoKettle nicht weiter verfolgt. Zwar wurde in Vautard (2012) ein Ansatz vorgestellt, um für einen Anwendungsfall eine Datenmodelltransformation zu INSPIRE (teilweise) erfolgreich durchzuführen, aber in diesem Ansatz wurden die Datensätze im proprietären Vektorformat Shapefile der Firma ESRI verwendet. Als Ausgabefunktion wurde hier *OGR Output* gewählt, welche jedoch eigentlich nur GML 2 (mit Einschränkungen GML 3.1.1) beinhaltet [OGR GML 2015].

Auch GeoKettle ist intern relational strukturiert. Die Daten werden in Tabellen zwischengespeichert und stehen ausschließlich in dieser Form zur weiteren Bearbeitung zur Verfügung. Das bedeutet, dass die Objektorientierung der Anwendungsschemata komplett aufgelöst wird. In den Zieldaten wird wieder eine scheinbar korrekte Objektorientierung erzeugt, vermutlich jedoch nicht für komplexere Strukturen, wie beispielsweise «geographicalName».

5.4 Ergebnisse für HUMBOLDT Alignment Editor

5.4.1 Transformationsansatz von HALE

In HALE werden die Abbildungen (wie auch in FME) auf Ebene der Anwendungsschemata definiert (horizontale Transformation). Es stehen zwei Panels zur Verfügung, um im *Schema Explorer* Quellschema (source schema) und Zielschema (target schema) einzuladen und anzeigen zu lassen. Die Schemata werden in einer Baumstruktur angezeigt, welche die Vererbungshierarchie der objektorientierten Schemata wiedergibt. Jeder Workspace handelt eine einzige undirektionale Transformation ab. Das Alignment der Abbildungen wird vorgenommen, indem Quellelement(e) und Zielelement in den Baumstrukturen markiert und via Auswahl einer Funktion aus der Bibliothek verknüpft werden. Das Alignment wird auch grafisch in einem Panel abgebildet und ist dort weiterhin manipulierbar. Durch Anzeige einer Live-Transformation können hinzugefügte Datensätze zur Laufzeit mit den temporär angelegten Abbildungen im Zielschema angezeigt werden. Sind in einem Schema mehrere FeatureTypes enthalten und sollen diese auf denselben FeatureType im Zielschema abgebildet werden, sollte pro FeatureType ein eigenes Alignment angefertigt werden. Würde ein Attribut des Zielschemas mehrfach belegt werden, beispielsweise durch multiple Quellattribute aus verschiedenen FeatureTypes, kann HALE nicht zwischen den Quellelementen unterscheiden und prozessiert strikt die erste angegebene Funktion dazu. Und zwar unabhängig davon, welcher FeatureType tatsächlich in den Quelldaten präsent ist. Die einzige Ausnahme bildet die Funktion *Assign (Bound)*, mit der man einen neuen Wert erzeugen kann, insofern ein (als Bedingung) festgelegter Quelltyp durch die Quelldaten repräsentiert wird.

Die Ausführung der Transformation erfolgt über eine vorgegebene Abfolge von fünf Steuerungsdialogen, in der die Einstellungen dazu getroffen werden: Angabe/Bestätigung des Quellpfades, Auswahl des Zielformates (z.B. GML (FeatureCollection)), Angabe des Zielpfades (lokale Speicherung) und der internen Validation, Auswahl einer Koordinatenreferenztransformation und XML/GML-Einstellungen.

Da auch HALE eine Desktop-Applikation ist, besteht nicht die Möglichkeit einer direkten Einbindung in eine Webschnittstelle. Die Transformation findet dementsprechend offline statt.

Sobald ein Datensatz eingelesen wurde, wird hinter den Quelltypen und Attributen die Anzahl der Instanzen angezeigt. Ebenso im Zielschema, nachdem eine semantische Relation erfolgreich angelegt wurde. Tabelle 5.2 gibt alle Funktionen von HALE mit einer kurzen Umschreibung wieder.

| Bezeichnung | Umschreibung |
|--------------------------------|--|
| Allgemeine Funktionen | |
| Retype | Retypisierung von Typen |
| Merge | Zusammenführen mehrerer Instanzen eines Quelltypen |
| Join | Zusammenführen mehrerer Instanzen verschiedener Quelltypen |
| Create | Erzeugen einer neuen (weiteren) Instanz |
| Rename | Übertragung und Umbenennung von Attributen |
| Assign | Zuweisung von Attributwerten unabhängig vom Quellschema |
| Date extraction | Extrahieren eines Datums aus einem String |
| Regex Analysis | Bearbeitung von Strings unter Einbeziehung von String-Patterns |
| Generate Unique Id | Erzeugen einer einzigartigen Id (Präfix + gml:id) |
| Classification | Mapping zwischen Attributwerten und Listen |
| Formatted String | Erzeugen eines Strings |
| Inline Transformation | Zeilenweise Transformation von Attributen |
| Assign (Bound) | Zuweisung von Attributwerten eines bestimmten Quelltyps |
| Geometrische Funktionen | |
| Ordinates to Point | Erzeugt Punkt aus X, Y und optional Z |
| Network Expansion | Erzeugt einen Puffer um eine Geometrie |
| Calculate Length | Berechnet die Länge von Geometrien |
| Calculate Area | Berechnet die Fläche von Geometrien |
| Centroid | Berechnet Mittelpunkt einer Geometrie |
| Compute Extent | Berechnet Hüllen um eine Geometrie |
| Aggregate | Legt Geometrien zusammen |
| Groovy Script | |
| Groovy Script | Anlegen von Skripten in Groovy |
| INSPIRE | |
| Inspire Identifier | Erzeugung eines Identifiers gemäß INSPIRE-Kodierung |
| Geographical Name | Erzeugung der Attribute des komplexen Typs Geographical Name |
| Sonstige | |
| Mathematical Expression | Durchführung einfacher Berechnungen (XPath) |
| Generate sequential ID | Mathematische Erzeugung einer ID |
| XPath expression | Nutzung eines vorgefertigten XPath-Ausdrucks (XSLT) |

Tab. 5.2 – HALE: Funktionsbibliothek (vgl. [Reitz und Templer 2011])

5.4.2 Transformation der Geometrie mit HALE

Die Funktionsbibliothek der verwendeten Version der Software enthält sieben Funktionen zur Bearbeitung der Geometrie. Mit diesen können auch einfache Erweiterungen durchgeführt werden (u.a. Aggregation, Hüllen erzeugen). HALE ist nicht in der Lage, das Koordinatenreferenzsystem der originären Anwendungsfälle automatisch auszulesen. Stattdessen muss dieses beim Einlesen des Datensatzes durch den Anwender manuell eingegeben werden. Während Ausführung der Transformation wird im einem Dialogfenster abgefragt, ob eine Reprojektierung des Koordinatensystems erfolgen soll. Die Auswahl wird in beiden Fällen allerdings nicht durch eine Listenauswahl unterstützt, so dass dem Anwender die URI Identifier vorab bekannt sein müssen. Die Abbildung der Geometrie kann in jedem Knoten innerhalb der Geometrie festgelegt werden. Für alle Knoten unterhalb der Abbildungsfestlegung mit Angabe der `«gml:id»` wird jedoch eine neue `«gml:id»` erzeugt, anstatt diese aus den originären Features übernommen. Eine Recherche in den Nutzerforen ergab den Hinweis, dass HALE alle Attribute der Geometrie schlicht ignoriert, unter der Annahme, sie wären nicht weiter von Bedeutung [HALE Forum 2015].

Eine BoundingBox zur Angabe der Umgebung der gesamten Geometrie, welche im erzeugten GML-Instanzdokument dem FeatureMember voransteht, wurde nicht automatisch erzeugt. Auch ein Versuch die BoundingBox mit der Funktion *Compute Extend (bounding box)* aus den originären Geometrien zu berechnen ist nur bedingt gelungen, da das Ergebnis immer (alle Optionen) 2-dimensional angegeben wird. Des Weiteren zeigte sich eine Abweichung von einigen Metern im Vergleich zur Berechnung mit FME, wobei das Berechnungsverfahren nicht ersichtlich ist.

Transformationsfälle I & II

Es wurde die Funktion *Rename* verwendet, um die Geometrie in das Zielschema zu übertragen. Die Prüfung der Resultate ergab, dass die Geometrien in beiden Fällen und allen Datensätzen korrekt übertragen und reprojeziert wurde. Abbildung 5.8 zeigt ein Beispiel einer erzeugten Geometrie.

```
-<ps:geometry>
  -<gml:MultiSurface gml:id="BLFD_BAUDENKMAELER_2745_BLFD_GEOMETRY" srsName="http://www.opengis.net/def/crs/EPSSG/0/4258">
    -<gml:surfaceMember>
      -<gml:Polygon gml:id="_bd10aa4f-947e-4a26-97ee-d3858062704e" srsName="EPSSG:4258">
        -<gml:exterior>
          -<gml:LinearRing>
            -<gml:posList>
              48.607862866573015 12.826005796285104 48.60795916841507 12.82594458214927 48.60793054778138 12.825853869977244 48.60
            </gml:posList>
          </gml:LinearRing>
        </gml:exterior>
      </gml:Polygon>
    </gml:surfaceMember>
  </gml:MultiSurface>
</ps:geometry>
```

Abb. 5.8 – HALE: Beispiel für die Ausgabe der Geometrie in den Transformationsfällen I & II

Transformationsfall III

Bei diesem Transformationsfall setzt sich die Geometrie aus der *TerrainIntersection* und den *BoundarySurfaces* zusammen (siehe Abschnitt 3.1.3). Im Anwendungsschema ist die

TerrainIntersection ein Attribut Building der BuildingParts und kann mit der Funktion *Rename* unproblematisch auf das identische Zielattribut übertragen werden. Die BoundarySurfaces gehören im Quellschema dem Attribut consistsOfBuildingPart an und sind (wie in Abschnitt 3.1.3 beschrieben) in die Unterklassen ClosureSurface, WallSurface, GroundSurface und RoofSurface gegliedert. HALE beschränkt die Auswahl der Funktionen zur Übertragung der BoundarySurfaces jedoch, so dass diese nur aggregiert werden können, was allerdings ausschließlich ein einzelnes Polygon zur Folge hat. Die Funktion *Rename* kommt deshalb nicht in Frage, weil damit nur ein Quellelement ausgewählt werden kann und *Assign* operiert ganz ohne Quell-elemente. Wird die Zuweisung mit einem Groovy Script erzwungen, resultiert dies in der Angabe mit MultiSurfaces mit Angabe der xlink:href ohne Koordinaten. Auch die Versuche mit weiteren Funktionen (z.B. *Merge*, *Groovy Script (Create)*) sind gescheitert, da dadurch eine zweite Building-Instanz generiert wird. Die letzte Option war, mit Groovy Script jede Surface einzeln auszulesen und neu nach den BuildingParts anzuordnen, was ebenfalls nicht möglich war. Obwohl man sich in der Funktion die Zugriffsmöglichkeiten auf die Instanzen und Werte in Groovy-Code angeben lassen kann, wurde dennoch ein nicht näher spezifizierter Fehler ausgegeben. Woran die Übertragung der Geometrie letztlich scheitert bleibt unklar. Es könnte u.a. ein Problem in den gegebenen Funktionen bei der Zuordnung der einzelnen CompositeSurfaces zu den BuildingParts vorstellbar sein. In der Arbeit von Popp (2015, S.23f.) wurde ein ähnliches Problem mit dem Tool FME aufgedeckt. Dort war die Übertragung der Geometrie ebenfalls problematisch aufgrund der Struktur im Quellmodell.

5.4.3 Transformation der Semantik mit HALE

Die Funktionsbibliothek enthält 19 weitere Funktionen zum Erzeugen von Inhalten. Datentypskontrollierungen erfolgen automatisch in den Datentyp des Zielattributs durch Verwendung der Funktion *Rename*. Zwischen folgenden geometrischen Datentypen kann die Funktion ebenfalls automatisch konvertieren (vice versa): LineString <-> MultiPoint, Polygon <-> MultiPoint, MultiPolygon <-> MultiPoint, Polygon <-> LineString, Polygon <-> MultiLineString, MultiPolygon <-> LineString. Zur Übertragung von Attributwerten werden u.a. die Funktionen *Rename*, *Formatted String* und *Date extraction* verwendet, zum Erzeugen neuer Werte u.a. die Funktionen *Assign*, *Assign (Bound)* und *Formatted String*. Es ist nicht möglich, mit den vorgegebenen Funktionen Strukturen wie Schleifen oder Bedingungen zu generieren. Dafür ist eine Funktion zur Erstellung eigener Skripte mit Groovy vorhanden (Groovy Script), womit weitreichendere Möglichkeiten bestehen, insofern der Anwender über die entsprechende Programmierkenntnis verfügt. HALE bietet außerdem vorgefertigte Codelists und zwei spezielle Funktionen zum Gebrauch für Transformation zu INSPIRE an. Zuzüglich können auch eigene Codelists zum Mapping zwischen Attributwerten oder Listen verwendet werden (Funktion: *Classification*). Mathematische Funktionen werden durch XPath⁶ ausgedrückt.

⁶Die XML Path Language (XPath) ist eine Sprache des W3C zur Abfrage und Auswertung von XML-Dokumenten [Wikipedia 2015d].

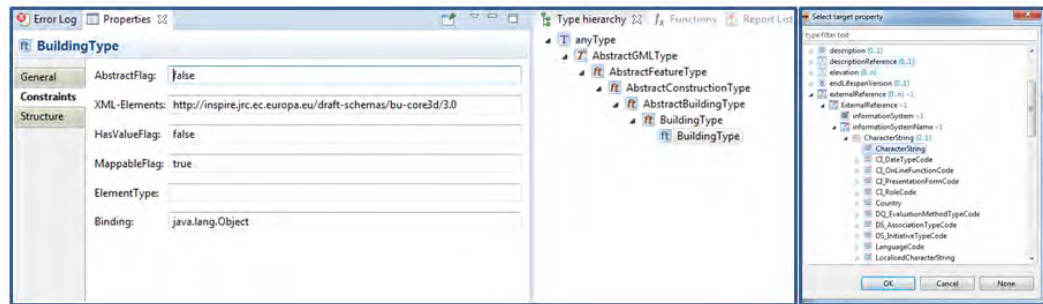


Abb. 5.9 – HALE: Anzeige von Attributen (rechts) und deren Eigenschaften (links)

Wie in Abbildung 5.9 rechts abgebildet, muss die Auswahl der Quell- und Zielelemente im korrekten Knoten erfolgen. Im Beispiel der Abbildung musste sogar der Datentyp des Attributs ausgewählt werden, da dieser in der Datenspezifikation [INSPIRE BU 2013, S.33] als freier Text deklariert wurde. In der Abbildung links werden die vorherrschenden Typhierarchien und die Eigenschaften des ausgewählten Quellelements gezeigt. Neben den allgemeinen Angaben zum Element (z.B. Namespace) und der Klasse, von der geerbt wird, können auch Constraints (Bedingungen) eingesehen und verändert werden (z.B. Multiplizität, Nillable-Flag, XML-Attribute-Flag). Falls Attribute keine Werte besitzen und `nillable="true"` sind, erzeugt HALE im Resultat automatisch die Angabe `xsi:nil="true"`.

Transformationsfall I & II

Die Zerlegung einzelner Attributwerte des Quellschemas ist nur durch Groovy Script möglich (siehe Abbildung 5.10). Darin wurden die Liste zweier Quellattribute ausgelesen, durch Angabe der Trennzeichen zerlegt und mit einem Character zu einem einzigen Zielattribut neu zusammengeführt. Im Gegensatz zu FME führt HALE die Belegung des Attributs `legalFoundationDocument` unproblematisch nach direkter Wertzuweisung aus. Das Attribut `designationLegalDefinition` kann noch nicht belegt werden, da es noch nicht Teil des Schemas ist. In Transformationsfall II werden im Quellschema sowohl die `id` als auch die `gml:id` gleichermaßen als `id` bezeichnet. Unterschieden werden können die Attribute, indem man sich die Attributwerte (bei geladenen Quelldaten) anzeigen lässt (zu finden in den Properties), oder die Eigenschaften näher untersucht, welche die `gml:id` als XML- und Pflichtelement auszeichnen. Nach Prüfung der Resultate zeigten sich keine weiteren Auffälligkeiten oder Fehler.

Transformationsfall III

Die Belegung der weiteren Inhalte erfolgte problemlos und analog zu den anderen beiden Transformationsfällen. Wobei für die Attribute des Stereotyps «voidable» auch hier nur mehr der `VoidReasonValue` erzeugt werden musste.

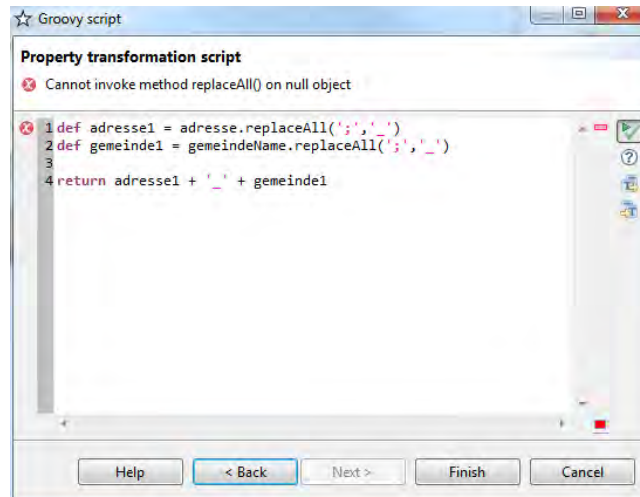


Abb. 5.10 – HALE: Erzeugung eines Attributwertes mit Groovy Script

5.4.4 Lösungsstrategien für die klassifizierten semantischen Relationen

Nachfolgend werden die gewählten Lösungsansätze zur Abbildung der semantischen Relationen in HALE knapp beschrieben und die verwendeten Funktionen angegeben:

- **SR.I.1 (DT): Äquivalentes Konzept im Quellmodell mit unterschiedlichem Datentyp**
HALE führt Datenkonvertierungen automatisch durch.
Rename
- **SR.I.2 (WZ): Äquivalentes Konzept im Quellmodell**
Schlichte Übertragung des Attributwertes.
Rename
- **SR.I.3 (WZ): Unterschiedliches Konzept im Quellmodell mit unterschiedlicher Strukturierung**
Auch in HALE können alle eingelesenen Attribute bei der Erzeugung eines Zielwertes mitwirken. Mit Groovy Script lassen sich die Attributwerte zerlegen, verknüpfen und manipulieren, wobei jedoch die Syntax beherrscht werden muss (siehe Abbildung 5.7).
Groovy Script
- **SR.I.4 (WZ): Fehlendes Konzept im Quellmodell**
Ist das Zielattribut «voidable», wird es im *Schema Explorer* ausgewählt und der Grund des Fehlens (z.B. “unknown”) angegeben.
Assign
- **SR.I.5 (WZ): Fehlendes Konzept im Quellmodell mit möglicher Herleitung**
Das Zielattribut wird ausgewählt und durch die Funktion mit einem Wert belegt.
Assign
- **SR.III.2 (WZ): Identisches Konzept im Quellmodell**
Die Wertbelegung erfolgt analog zu SR.I.2 durch Übertragung des Attributwertes ohne weitere

Eigenschaften beachten zu müssen.

Rename

- **SR.III.5 (WZ): Kompatibles Konzept im Quellmodell mit unterschiedlichen Modellierungskonstrukten**

Es konnte keine geeignete Funktion identifiziert werden. Der Grad der geometrischen Heterogenität scheint HALE Probleme zu bereiten, möglicherweise aufgrund dessen, dass mehrere Quellattribute in Form von `CompositeSurfaces` an der Bildung der Geometrie beteiligt sind.

5.4.5 Performance von HALE

Während des Einlesens großer Datensätze (z.B. Baudenkmäler mit allen 143.219 darin enthaltenen Features) zeigte sich ein Anstieg der Last im Arbeitsspeicher [KB] um bis zu 54% von 577.050 KB auf 1.056.825 KB. Da die Auslastung dann konstant blieb und nach Entfernen der Daten wieder auf den ursprünglichen Wert absank, deutet dies darauf hin, dass die Daten zur Laufzeit permanent im Arbeitsspeicher gehalten werden. Bei der Durchführung der Transformation sollten deshalb alle nicht benötigten Prozesse unbedingt beendet werden, um eine Instabilität des Systems zu verhindern. Die exakte Laufzeit der Transformationen konnte nicht ermittelt werden, da im Report nur ein Zeitstempel zur Beendigung des Vorgangs angegeben wird.

5.4.6 Verwendung von HALE abseits von INSPIRE

HALE wurde speziell für die Anwendung mit INSPIRE entwickelt. Da eine Skriptsprache in HALE integriert wurde, dürften die Möglichkeiten aber auch für andere Anwendungsfälle unter Einschränkungen ausreichend gegeben sein. Für komplexere Anwendungsfälle, insbesondere auch bei der Übertragung komplexer Geometrien, könnte die Transformation mit HALE schwierig werden, da die Funktionsbibliothek für die Anwendung mit INSPIRE ausgelegt ist. Die Auswahl der verarbeitbaren Formate ist jedoch deutlich eingeschränkter als beispielsweise bei FME. Dem Anwender müssen außerdem stets die Anwendungsschemata vorliegen, da diese nicht in HALE intern hinterlegt sind.

5.4.7 Berichtswesen und Validierung in HALE

HALE generiert bei der Transformationsausführung einen Report, der als Textdokument gespeichert werden kann. Fehlermeldungen werden leider nicht weiter erläutert, sondern nur erwähnt. Zur Validierung kann eine Option in einem der Dialogfenster nach Start der Transformation bestätigt werden. Zu allen durchgeführten Transformationen aller Fälle ergab die Validierung ein negatives Ergebnis, ohne Angabe von Gründen und obwohl die Zieldokumente durch externe Programme (u.a. XML-Editoren) als valide eingestuft wurden. Die Tauglichkeit des Mechanismus ist also eindeutig zu hinterfragen.

5.5 Vergleich der Ergebnisse

FME lieferte für die beiden ausgewählten Anwendungsfälle der Landesämter vollständig valide und wohlgeformte Ergebnisse. GeoKettle produzierte unbrauchbare Ergebnisse. Demzufolge sind auch im digitalen Anhang keine Datensätze für GeoKettle enthalten. Für die Transformationsfälle I & II konnte HALE (fast) fehlerfreie Datensätze generieren (mit Ausnahme, dass die `gml:id` nicht in jedem Knoten übernommen wurde). Im Transformationsfall III konnte HALE die Geometrie nicht korrekt erzeugen, wohingegen FME nicht zur Anwendung kam. Aus den Erkenntnissen der Arbeit von Popp (2015) kann jedoch gefolgert werden, dass nach derzeitigem Entwicklungsstand von FME und HALE, FME das geeignetere ETL-Tool ist, um CityGML-Daten zu transformieren. Möglicherweise ist die Abbildung der Geometrien jedoch im Anwendungsschema `BuildingsExtended3DLoD2` durchaus möglich, da darin auch die Gebäudehüllen mit den `boundary Surfaces` (z.B. `Wallsurfaces` etc.) gebildet werden und die `Surfaces` somit einzeln abgebildet werden können [DS BU 2013, S.108f.], mit der Funktion *Rename*.

Kapitel 6

Bewertung und Vergleich der Transformationswerkzeuge

In diesem Kapitel greift die vorliegende Arbeit die Ergebnisse (Kapitel 5) und Erkenntnisse des Praxis- teils auf und geht auf Grundlage dessen zur Bewertung der Datenmodelltransformationstools und deren Vergleich über. Aus den Erkenntnissen wurden auch Problemstellungen von Datenmodell- transformationen und ihrer Darstellung in Abbildungstabellen ermittelt und werden nun nachfolgend erläutert.

6.1 Problemstellungen

Obwohl sich die Umsetzung der Transformationen in den jeweiligen ETL-Tools auf die Abbildungsta- bellen stützen kann und somit ein großer und anspruchsvoller Teil der semantischen Datenmodell- transformation bereits exemplarisch verwirklicht wurde, bestehen durchaus noch zahlreiche Problem- stellungen, die es zu überwinden gilt. Deutlich wurde dies u.a. auch in der Arbeit von Popp (2015), in der aufgedeckt wurde, dass Abbildungstabellen, obgleich Hilfestellung, auch eine Fehlerquelle dar- stellen können. Bei der Implementierung der Transformationsabbildungen in einem ETL-Tool sollten verwendete Quellen deshalb kritisch hinterfragt und im Zweifel überprüft werden.

6.1.1 Eigenheiten der Datenmodelltransformationstools

Ein ETL-Tool muss ein Datenmodell bzw. einen zu transformierenden Datensatz zunächst einmal einlesen, um die darin enthaltenen Daten weiterverarbeiten zu können. Die gewählten Anwendungs- schemata liegen, wie bereits erläutert (siehe u.a. Abschnitt 4.1), im objektorientierten Paradigma vor. Das bedeutet, dass die Quelldaten - und später auch die Zieldaten, wie gefordert - in einer ob- jektorientierten Hierarchie angeordnet sind. Möglicherweise arbeitet ein Tool jedoch intern mit einer relationalen Strukturierung in tabellarischer Form. In diesen Fällen muss durch das Tool gewährleis- tet werden, die Objektorientierung wieder korrekt reproduzieren zu können.

Manche Tools verfügen dadurch über eine Spezialisierung für bestimmte Anwendungsfälle, dass programmatische Inhalte bereits implementiert sind, wie beispielsweise Schemata (z.B. in dem Tool

FME) oder Kodierungslisten (z.B. in dem Tool HALE). Wird auf diese Programminhalte zurückgegriffen, ist jedoch sicher zu stellen, dass sie inhaltlich und syntaktisch korrekt sind und in der aktuellen Version vorliegen.

In Popp (2015) trat der Fall in Erscheinung, dass beim Einlesen der Daten (CityGML) mit FME die Geometrie durch die Software modifiziert/abgeändert wurde, denn FME überträgt die Geometrie zunächst in ein internes Modell (siehe Abschnitt 5.2.5). Es musste demzufolge eine Verarbeitungsstruktur implementiert werden, welche die Geometrie wieder in die gewünschte Form zu bringen vermag. Anscheinend war dies nach den Erkenntnissen von Popp (2015) nicht der Fall. Auch HALE konnte gemäß den Ergebnissen der vorliegenden Arbeit die Geometrie in einem Fall nicht verwerten, obwohl durch HALE keine interne Umstrukturierung vonstatten geht.

Es unterscheiden sich auch die Vorgehensweisen bei der Initiierung und Durchführung von Transformationen, was in unterschiedlichen Vor- und Nachteilen resultiert. Diese Vor- und Nachteile spiegeln sich in den Bewertungen diverser Kriterien wieder (siehe Abschnitt 6.2).

6.1.2 Grenzen der Datenmodelltransformation

Die hier vorgestellten Datenmodelltransformationen können anhand von Unterscheidungsmerkmalen näher klassifiziert werden. In Fichtinger (2011, S.50ff.) wurden Ansätze zur Klassifizierung mit Unterscheidungsmerkmalen aus diversen Forschungsarbeiten aufgegriffen und zu einem einheitlichen Schema zusammengefasst. Tabelle 6.1 klassifiziert die vorliegenden Transformationsfälle nach diesem Schema. Da die Klassifizierung, unterschieden nach den Transformationsfällen, identisch ausfallen würde, wird sie nach ETL-Tool unterschieden vorgenommen.

Aus der Klassifizierung der vorliegenden Anwendungsfälle (siehe Abschnitt 3.1.4) lassen sich Rückschlüsse in Bezug auf die Grenzen von Datenmodelltransformationen ziehen, insbesondere die Anwendungsfälle betreffend. Die Erkenntnisse lassen sich aber auch aus dem Speziellen auf das Allgemeine übertragen. Da für die Quellmodelle Denkmäler und Schutzgebiete keine Dokumentation zur Verfügung steht, können verwendete Kodierungsregeln nur auf Basis der Instanzen vermutet werden, oder müssen beim Datenanbieter erfragt werden. Andererseits stellt die hier praktizierte Vorgehensweise, in Ermangelung einer UML-Modellierung, die GML-Anwendungsschemata direkt aus den vorhandenen Shapefiles mit einem WFS-Service zu erzeugen, eine praktikable Alternative dar [Fichtinger 2011, S.141f.]. Laut Kutzner und Eisenhut (2010, S.60f.) und Fichtinger (2011, S.142) geschieht dies jedoch nur durch Einbußen der Plattformunabhängigkeit.

Aus den Beispielen der Anwendungsfälle und auch aus der Klassifikation der Transformationsansätze (Tabelle 6.1) lassen sich noch weitere Punkte ableiten, die der Machbarkeit von Datenmodell-

| Unterscheidungs- merkmal | FME | GeoKettle | HALE |
|--|---|---|---|
| Richtung der Transformation | horizontal | horizontal | horizontal |
| Ebene der Abbildung | Transferformat- schemata (XSD) | Transferformat schemata (XSD) | Transferformat schemata (XSD) |
| Transformations- sprachenparadigma | operational & deklarativ | keines | EDOAL/gOML (deklarativ) |
| Repräsentation der Transformationssprache | textuell | textuell | textuell |
| Automatisierungsgrad d. Abbildung | unterstützt durch GUI & teil-automatisiert | unterstützt durch GUI & strikt manuell | unterstützt durch GUI & teil-automatisiert |
| Systemabhängigkeit | proprietär | systemunabhängig | systemunabhängig |
| Umkehrbarkeit | nicht gegeben | nicht gegeben | nicht gegeben |
| Ausführung der Transformationen | offline online möglich | offline | offline online möglich |

Tab. 6.1 – Klassifizierung der Transformationsansätze

transformationen Grenzen setzen und im folgenden aufgegriffen werden:

- Eine entscheidende Grenze aus technischer Sicht bildet der aktuelle Stand der Technik. Nicht für alle in Frage kommenden Formate ist die Entwicklung hinsichtlich der Interoperabilität in dem Maß fortgeschritten, dass beliebig zwischen Modellen transferiert werden kann. An dieser Stelle sei allein angeführt, dass beispielsweise laut Kutzner und Eisenhut (2010, S.67) die Abbildungssprache RIF nicht geeignet ist, objektorientierte Strukturen zu verarbeiten (siehe Abschnitt 2.4.2), aber andererseits von Beare et al. (2010, S.20, S.65) festgestellt wird, RIF würde anscheinend die Ziele des INSPIRE Transformation Network Service¹ sehr genau erfüllen. Folglich wäre es denkbar, dass bei tatsächlicher Verwendung von RIF objektorientierte Strukturen von Daten verletzt werden und somit keine Interoperabilität bei der Transformation besteht. Andere Formate vermögen Objektorientierung zu handhaben, sind aber womöglich aus anderen Gründen weniger gut zur Realisierung des INSPIRE-Dienstes geeignet. Das eben genannte Beispiel zeigt auch die starke Kapselung bzw. Spezialisierung von Formaten. Beispielsweise können die in HALE definierten Transformationsregeln auch im Format CSV exportiert werden. Da darin jedoch auf Funktionen von HALE verwiesen wird (z.B. *Rename*), sind diese Transformationsregeln für ein anderes ETL-Tool nicht zu gebrauchen.
- Auch Transformationswerkzeuge sind aus verschiedenen Aspekten heraus unterschiedlich gut geeignet. Jedoch beschränken sich die dargelegten Ergebnisse und Erkenntnisse dieser Arbeit vor allem auf die vorliegenden Transformationsfälle. Treten komplexere Fälle in Erscheinung,

¹Der INSPIRE Transformation Network Service (INSPIRE TNS) ist ein Dienst (in der Regel online) zur Transformation von Gedaten im Hinblick auf die Verwirklichung von Interoperabilität [Beare et al. 2010, S.65].

muss in diesen Einzelfällen eine festgestellte Eignung gegebenenfalls revidiert werden.

- Betroffen sind vor allem auch Fachanwendungen (wie z.B. die hier diskutierten Anwendungsfälle), die eine spezifische Wahrnehmung von Objekten eines Realweltausschnitts und folglich eine eigene Fachsemantik bedingen. Doch auch Fachanwendungen, die einander bislang fremden Anwendungsbereichen entstammen, werden im Zuge der voranschreitenden Digitalisierung in zunehmendem Maße zu neuen Anwendungen in einer gemeinsamen Domäne zusammengeführt. Werden allerdings Transformationen benötigt, muss womöglich aus u.a. ontologischen Gründen² erst eine semantische Modifizierung der Modelle vorgenommen werden. Ein guter Ansatz ist die Weiterentwicklung von ontologischen Ansätzen, die auch zur Förderung von Interoperabilität beitragen können. In Zedlitz (2013) wurde ein Ansatz erarbeitet, um die - auf Ontologien basierte - Abbildungssprache OWL³ zu UML und vice versa transferieren zu können. Dieser Ansatz könnte beispielsweise dazu dienen, zwischen heterogenen konzeptuellen Modellierungen auf Basis einer gemeinsamen Ontologie-Domäne zu vermitteln.
- Während der Arbeit von Fichtinger (2011) wurde bei Verwendung eines INSPIRE betreffenden Transformationsfalls bemerkt, dass nicht im Detail explizit modellierte Eigenschaften auftreten. Dies wurde bei Betrachtung der Modellierung der Geometrie offenkundig, die im Profil lediglich als „abstrakte Basisklasse `GM_Object` als Geometrietyp angegeben“ [Fichtinger 2011, S.116] ist. Das Anwendungsschema Protected Sites ist ebenfalls davon betroffen. Da „alle Subtypen von `GM_Object` zulässig“ [Fichtinger 2011, S.116] wären, könnte der Fall auftreten, dass in genau einem der beiden Modelle die Geometrie explizit durch einen der Subtypen festgelegt wird, allerdings bei der Transformation fälschlicherweise ein anderer Subtyp instanziiert wird. Staub (2009, S.142f.) bezeichnet einen ähnlichen Sachverhalt in übergeordnetem Kontext als „Zwang zur klaren Profilierung“. Insbesondere bei bidirektionalen Transformationen stellt dies eine Unsicherheit dar, wenn diese Ungenauigkeiten zur Änderung der Semantik führen, oder eine falsche Syntax bewirken. In der Schlussfolgerung kann deshalb festgehalten werden, dass mangelnde Detaillierung in den Modellen der Korrektheit von Modelltransformationen Grenzen setzt.
- Die hier vorliegenden Anwendungsfälle haben aufgezeigt, dass nicht immer eine konzeptuelle Modellierung vorliegt, oder greifbar ist (z.B., da sie nicht veröffentlicht wurde). Folglich schließt dieser Sachverhalt auch aus, Transformationen auf Ebene der konzeptuellen Modelle durchführen zu können. Somit kann ein Anwender auch nicht Nutznießer der dadurch erzielten Vorteile sein. Eines dieser Vorteile wird durch Plattformunabhängigkeit gewonnen, die laut Fichtinger (2011, S.140) dazu genutzt werden kann, dass „theoretisch beliebiger, für die Ausführung der Transformation in unterschiedlichen Software-Werkzeugen benötigter Code abgeleitet werden“

²Beispielsweise, da unterschiedliche Fachanwendungen äquivalente Realweltausschnitte und eventuell auch identische Objekte differenziert auffassen/modellieren und Bezeichnungen üblicherweise an die Fachsprachen angelehnt sind.

³Die **Web Ontology Language (OWL)** ist eine Sprache zur Definition und Übertragung von Inhalten [Zedlitz 2013]; gemeint ist OWL 2.

kann (vertikale Transformation). Dieser Aspekt fällt auch zu Lasten der Automatisierbarkeit von Transformationen. Da in den vorliegenden Transformationsfällen I / II auf eine konzeptuelle Modellierung verzichtet wurde, wird auch in diesen Fällen die Interoperabilität zwischen Modellen und Plattformen eingeschränkt.

- Wie es aus den hier vorliegenden Transformationsfällen I & II ersichtlich wird, besteht hier eine gewisse Diskrepanz im Umfang und in der Detaillierung zwischen Quell- und Zielmodellen. Eine Reihe von Attributen, die in den Quellmodellen der Landesämter geführt werden, wurden nicht in Zielattributen abgebildet, stattdessen entfallen sie schlichtweg. Im Quellmodell Denkmal BLdF existiert beispielsweise eine ausführliche Bezeichnung des Objektes in Form einer Beschreibung der Objektinstanzen, für die kein alternatives Zielattribut identifiziert wurde. Die Attributwerte gehen also im Zuge der Transformation verloren. Würde eine Rücktransformation erfolgen (dann läge eine bidirektionale Transformation vor), könnten die ursprünglichen Attributwerte auch nicht mehr rekonstruiert werden, was zwei inhomogene Instanzen im originären Modell zur Folge hätte.
- Zu große Datenmengen können eine Transformation ebenfalls sabotieren, insbesondere wenn eine online durchgeführte Transformation innerhalb eines Netzdienstes angedacht ist. Zwar kann durch Portionierung der Datensätze die Performance erhöht werden, dennoch ist dann die Durchführung der vollständigen Transformation mit mehr Aufwand verbunden (z.B. mehr Personalstunden).
- Für diese Arbeit wurden als Eingangsformate GML-Anwendungsschemata bzw. CityGML gewählt. Staub (2009, S.141) beschreibt einen Fall der aufzeigt, dass in Transferformaten nicht die komplette Semantik eines Datenmodells repräsentiert werden könne. Demzufolge würden in Interlis⁴ semantisch unterschiedliche Geometrietypen im Transferformat als gleiche Objekte kodiert werden. Transferformate stellen also nicht unbedingt die optimale Grundlage für Modelltransformationen dar, wenngleich sie auch aus u.a. praktikablen Gründen gern verwendet werden.

6.1.3 Grenzen der Darstellung in Abbildungstabellen

Bislang bildete die Aufbereitung der semantischen Elementrelationen in Abbildungstabellen eine praktikable Grundlage bei der Projektabwicklung von Datenmodelltransformationen, da sie eine plattformunabhängige Wiederverwendung ermöglichen. Die eigentliche Stärke der tabellarischen Form liegt darin, die Schemata gegenüberstellen zu können und somit eine bessere Lesbarkeit der Abbildungen zu generieren. Das bedeutet, dass anhand des zeilen- und spaltenweisen Aufbaus, die

⁴Interlis ist eine in der Schweiz verwendete Modellierungssprache zur Beschreibung konzeptueller Schemata.

semantischen Zusammenhänge zwischen den Schemata abgelesen werden können. Die Zeilen repräsentieren somit gewissermaßen einen gedachten Fremdschlüssel zwischen den Schemaelementen. Bei näherer Betrachtung sind der Darstellung in Abbildungstabellen jedoch ebenfalls Grenzen gesetzt:

- Es wird durch eine einzigartige Zelle die semantische Relation festgelegt, in der Form, dass innerhalb einer Zeile, sich die abzubildenden Elemente der Schemata gegenüberstehen und in der Zelle die benötigte Transformationsfunktion angegeben wird. Dabei kann jedoch nur eine der beiden Schemastrukturen geordnet abgebildet werden, da die Schemata nicht für diesen Zweck ausgelegt sind. Würden beide Strukturen gemäß ihrer Hierarchien angeordnet werden, befänden sich die in Relation zu setzenden Elemente vermutlich nicht innerhalb derselben Zeile, oder nur manche davon. In der Regel wird die Struktur des Zielschemas zeilenweise angeordnet und die Struktur des Quellschemas aufgelöst.

Eine Ausnahme bildet die Mapping-Tabelle zu Transformationsfall III, in der gleich vier Anwendungsschemata enthalten sind. Jedoch funktioniert dort die Anordnung problemlos, da die Anwendungsschemata stark homogen zueinander sind und aufgrund fehlender Attribute nur vereinzelt Lücken entstehen. Das bedeutet, dass an der Stelle eines fehlenden Attributs eine Zeile in Partition eines Schemas leer bleibt, aber die Struktur der Tabelle durch kleine Inhomogenitäten in keinsten Weise gestört wird.

- Grundsätzlich kann auch der Fall abgebildet werden, bei dem mehrere Quellelemente eine semantische Relation zu einem Zielelement bilden ($n:1$). Optional können dann in der entsprechenden Zelle zur Angabe des Quellattributs alle beteiligten Attribute dieses Schemas gelistet werden und/oder die Attribute sind in der Repräsentation⁵ der Transformationsfunktion enthalten. Der umgekehrte Fall, dass ein Quellattribut zur Bildung von Relationen mit mehreren Zielattributen verwendet wird ($1:n$), ist anordnungsbedingt nur dadurch darstellbar, dass die Quellelemente redundant genannt werden. In diesen Fällen würde es sich anbieten, Spalten hinzuzufügen, die tatsächliche Fremdschlüssel in textueller Form beinhalten. Anstatt über die Zeilen könnten alternativ über diese die Zuordnungen stattfinden. Jedoch ist dann der eigentliche Sinn der Abbildungstabellen nicht mehr gegeben. Sehr kompliziert und nicht mehr darstellbar sind Fälle, in denen mehrere Quellattribute auf mehrere Zielattribute abgebildet werden ($n:m$), da die Anordnung in Tabellen die Darstellung dieser Form von semantischen Relationen kaum noch zulässt.
- Es ist in den Abbildungstabellen in der Regel auch eine Richtung der Transformation hinterlegt, indem Ziel- und Quellschema festgelegt werden (s.u.). Innerhalb einer Abbildungstabelle mit schwach homogenen bis heterogenen Anwendungsschemata kann nur eine Transformationsrichtung dargestellt werden. Die entgegengesetzte Transformationsrichtung stellt eine eigen-

⁵Aufgrund der Plattformunabhängigkeit werden Transformationen als Pseudocode angegeben.

ständige Tabelle dar, da nun das vormalige Quellschema strukturbestimmend ist. Bidirektionale Abbildungen sind bei Verwendung stark homogener Anwendungsschemata (wie der Mapping-Tabelle zu Transformationsfall III) gestaltbar.

- In den Abbildungstabellen werden insbesondere auch die Objekttypen gegenübergestellt. Während in den vorliegenden Transformationsfällen die Objekttypen der Schemata rar sind (I & II) oder stark homogen (III), sind auch komplexere Anwendungsfälle denkbar. Möglicherweise könnten semantische Relationen stark verschachtelt abgebildet werden, so dass die Objekttypen mehrfach horizontal respektive vertikal angeordnet sein müssen. Die Komplexität in den Relationen ergibt sich dabei durch Anzahl der beteiligten Elemente an der Bildung von semantischen Relationen. Beispielsweise könnte ein Element eines Quellobjekttypen in die Bildung mehrerer Elemente verschiedener Zielobjekttypen einfließen (1:n), wie oben bereits erwähnt. Objekttypen des Zielschemas müssen vertikal untereinander stehen, da in horizontaler Anordnung die semantische Relation nicht mehr eindeutig wäre. Um zu verhindern, dass eine chaotische Anordnungen die Struktur der Tabelle zerstört, sind wiederum die Quellobjekttypen redundant aufzulisten. In der Folge würden die Abbildungstabellen sehr groß und unübersichtlich werden.
- Durch den Bereitsteller von Abbildungstabellen muss gewährleistet werden, dass die darin enthaltenen semantischen Relationen und Funktionen zur Übertragung vom Quell- in das Zielschema, trotz plattformunabhängiger Ausdrucksweise eindeutig zu interpretieren ist. Die Ausdrücke müssen deshalb so neutral wie möglich und so eindeutig wie möglich gewählt werden.

6.2 Festlegung der Bewertungskriterien

6.2.1 Kriterien zur Evaluierung der Ergebnisse

Manche Programme stellen auch Funktionalitäten zur Qualitätskontrolle der durchgeführten Prozesse zur Verfügung. Diese können Möglichkeiten beinhalten, die erzeugten Daten gegen ihr Zielschema zu validieren oder in - zur Laufzeit erzeugten - Dokumenten über den Transformationsvorgang zu berichten, auf Fehler oder Auffälligkeiten hinzuweisen und Kennwerte dazu festzuhalten (z.B. die benötigte Zeit). Dennoch müssen die Ergebnisse extern auf Korrektheit geprüft werden, da die Programm-interne Qualitätsprüfung zu den Transformationen ebenfalls Teil des zu bewertenden Leistungsspektrums der Werkzeuge ist. Auch geben diese Funktionalitäten keine Auskunft über interne Prozesse. Eine vollständige Prüfung der Korrektheit der erzielten Resultate aus den Transformationsfällen I bis III ist Gegenstand dieser Arbeit. Nachfolgend wird ein Schema bestehend aus Teilkriterien entworfen und in einem Katalog kurz charakterisiert. Zu jedem Kriterium ist eine eigenständige Wertung möglich. Dazu wird für jedes Kriterium ein eigener Bewertungsschlüssel angegeben. Aus den

Einzelwertungen lässt sich dann eine Gesamtwertung zur Evaluierung vornehmen.

Zur Evaluierung der erzielten Ergebnisse hinsichtlich ihrer Korrektheit werden folgende Parameter identifiziert und in Bewertungskatalog A zusammengefasst: Evaluierung der Ergebnisse.

- **A.0 Gesamtwertung der Evaluierung der Ergebnisse**

Die Gesamtwertung der Kriterien stellt die abschließende Evaluierung der Resultate dar.

0: Evaluierung stellt bedingt durch Fehler ungenügende oder mangelhafte Resultate fest

1: Evaluierung stellt korrekte Resultate fest

- **A.1 Prüfung auf Wohlgeformtheit zum Zielmodell**

Es muss festgestellt werden, ob die Zieldateien alle benötigten Strukturen aufweist (z.B. Open- und Closetags) sowie Regeln beachtet und somit hinsichtlich des XML-Paradigmas gültig ist.

0: Daten sind nicht wohlgeformt

1: Daten sind wohlgeformt

- **A.2 Prüfung auf Validierung zum Zielmodell**

Die Daten sind dann als valide anzusehen, wenn sie in Syntax und Semantik dem zugrundeliegenden Schema entsprechen.

0: Daten sind nicht valide

1: Daten sind valide

- **A.3 Prüfung auf semantische Vollständigkeit**

Es ist zu prüfen, ob auch tatsächlich alle definierten Abbildungen bei der Transformation durchgeführt wurden und die Semantik hinsichtlich des Zielschemas vollständig ist.

0: Daten sind semantisch nicht vollständig

1: Daten sind semantisch vollständig

- **A.4 Prüfung der geometrischen Transformation**

Die ursprüngliche Geometrie eines Objektes muss bei der Transformation erhalten bleiben.

0: Geometrie ist nicht korrekt

1: Geometrie ist korrekt

- **A.5 Prüfung der Koordinatenreferenz-Transformation**

Die Geometrie muss im korrekten Koordinatenreferenzsystem an identischem topologischen Ort referenziert sein.

0: Koordinatenreferenz ist nicht korrekt

1: Koordinatenreferenz ist korrekt

- **A.6 Prüfung der Identifikator-Zuweisung**

Es muss sich um einen einmaligen, persistenten und wieder erzeugbaren Objektidentifikator handeln.

0: Objektidentifikatoren sind nicht korrekt

1: Objektidentifikatoren sind korrekt

- **A.7 Prüfung der Kodierung**

Es ist festzustellen, ob die vorgesehene Kodierung verwendet wurde und zur Laufzeit erzeugte Elemente den Kodierungsregeln entsprechen.

0: Kodierung ist nicht korrekt

1: Kodierung ist korrekt

- **A.8 Prüfung der Objekthierarchien**

Die erzeugten Daten müssen die im Ziel-Anwendungsschema festgelegten objektorientierten Strukturen korrekt wiedergeben.

0: Objektorientierung ist nicht korrekt

1: Objektorientierung ist korrekt

Idealerweise sollten die erzielten Daten vollumfänglich korrekt erzeugt worden sein. Allerdings wird dass nicht für jeden Transformationsfall bzw. nicht für jedes verwendete ETL-Tool der Fall sein und zudem besteht auch immer die Möglichkeit zu Ausnahmen im Einzelfall. Der evaluierende Wertebereich, mit dem eine abschließende Aussage in Kategorie A.0 über die Einhaltung der Parameter getroffen wird, kann deshalb auch wie folgt definiert werden:

Jeder der Parameter kann (1) vollständig (100% korrekt), (2) größtenteils (vereinzelte Fehler), (3) überwiegend (viele Fehler), (4) ausgeglichen (in etwa gleichermaßen fehlerhaft sowie korrekt), (5) teilweise (häufiger fehlerhaft als korrekt) und (6) nicht (kaum bis gar nicht korrekt) erfüllt sein.

Anschließend kann noch unterschieden werden, in welchem Ausmaß, mit welchen Mitteln und in welchem Umfang Korrekturmaßnahmen vorgenommen werden müssen. Können beispielsweise Korrekturmaßnahmen innerhalb der Softwareumgebung getroffen werden, da sich möglicherweise Fehlerquellen eingeschlichen haben? Oder müssen externe Programme zur Lösung heran gezogen werden? Möglicherweise ist ein Fehler auch überhaupt nicht behebbar oder nur durch manuelle Manipulation der Zieldatensätze.

Ein entscheidender Faktor, der zur Evaluierung der Ergebnisse beiträgt, ist die Tatsache, ob die vorhandenen Fehler einen Datensatz für die Weiterverwendung unbrauchbar machen, oder ob er dennoch verwendbar ist. Und falls ja, ob die Nutzung der fehlerbehafteten Daten Einschränkungen zur Folge hat.

6.2.2 Kriterien zur Beurteilung der Datenmodelltransformationswerkzeuge

Die im vorherigen Abschnitt 6.2.1 behandelte Korrektheit der Transformationsergebnisse stellt die elementarste Bedingung bei der Beurteilung der Eignung von ETL-Tools dar. Liefert das Tool falsche, oder aufgrund von Fehlern unbrauchbare Ergebnisse, so ist es schlichtweg ungeeignet. Eine Ausnahme ist lediglich gegeben, falls eine ansonsten geeignete Software Fehlerquellen aufweist, wegen derer die Konformität der resultierenden Daten zum Zielmodell sabotiert wird und die Fehler mit einem Softwareupdate leicht behoben werden können. Sind die Fehler jedoch gravierender, so ist

das Tool nach einem Update nochmals in der neuen Version zu prüfen. Es ist auch fraglich, wie schnell dann ein Update realisiert und online gestellt werden kann. Dies hängt auch mit dem Grad der Professionalität und den Ressourcen des Herstellers zusammen, oder aber auch mit einer aktiven Nutzer-Community, die ebenfalls Lösungsansätze generieren kann. In jedem Fall sind gewisse Zusammenhänge zu erkennen, die in einem Anforderungsprofil an ein ETL-Tool wechselwirken und deshalb auch bei der Entwicklung des Bewertungsschemas beachtet werden sollen. Zunächst werden jedoch die identifizierten Rubriken im Detail vorgestellt, welche zur Bewertung gebildet wurden. Jede Rubrik (im folgenden Gruppe genannt) repräsentiert einen Anforderungsbereich an das Tool und beinhaltet ihm untergeordnete Bewertungskategorien mit Kriterien, die bei Erfüllung zur erfolgreichen Evaluierung in dem Anforderungsbereich beitragen. Die Anforderungsbereiche wurden teilweise schon in Fichtinger (2011, S.67f.) identifiziert und beschrieben, ihrerseits auch unter Verwendung weiterer Quellen. Die in Fichtinger (2011, S.50f.) erstellte Klassifizierung der Transformationsansätze fließen ebenfalls mit ein. An dieser Stelle erfolgt eine Weiterentwicklung dieses Ansatzes. Insgesamt werden im Folgenden in den Bewertungskatalogen B / C 51 einzelne Kriterien identifiziert und vorgestellt. Für jedes Kriterium wird wiederum ein eigener Bewertungsschlüssel festgelegt und kurz beschrieben.

Bewertungskatalog B: Leistungsvermögen der Software.

Kategorie B.1: Transformationsansatz der Software

Bewertungskategorie B.1 bündelt die Kriterien zur Beschreibung und Klassifizierung des Ansatzes und der Form von durchführbaren Datenmodelltransformationen mit der Software.

- **B.1.0 Gesamtwertung des Transformationsansatzes**

Trifft eine Gesamtwertung zum Ansatz der Datenmodelltransformation unter Berücksichtigung aller Teilkriterien dieser Bewertungskategorie.

0: Transformationsansatz ist ungenügend oder mangelhaft

1: Transformationsansatz ist ausreichend

- **B.1.1 Systemabhängigkeit**

Gibt die Abhängigkeit des Tools von einer Plattform an.

0: Systemabhängigkeit ist gegeben (proprietär)

1: Systemabhängigkeit ist nicht gegeben (Open-Source)

- **B.1.2 Mögliche Abbildungsebenen**

Gibt mögliche Ebenen zur Definition und Anwendung der Abbildung gemäß der *OMG 4-Schichten-Architektur* an: Datenebene M0 oder Transferformatebene M1 (siehe Abschnitt 2.1.3).

0: Datenebene (Abbildung und Anwendung)

1: Transferformatebene (nur Abbildung)

2: Transferformatebene (Abbildung) und Datenebene (Anwendung)

- **B.1.3 Initialisierbare Richtungen der Transformation**

Gibt mögliche Richtungen der Transformation an (siehe Abschnitt 2.4.1). (ja/nein)

i) undirektional ii) bidirektional iii) horizontal iv) vertikal

- **B.1.4 Arten der möglichen Transformationsausführung**

Gibt die Möglichkeiten der Ausführung der Transformation an (siehe Abschnitt 2.4.1). (ja/nein)

i) offline ii) on-the-fly

- **B.1.5 Skalierbarkeit der Transformationen**

Klassifiziert die Performance der Software in Abhängigkeit der Datenmenge. Relative Angaben zur Skalierbarkeit ohne Wertung.

- **B.1.6 Mögliche Strukturen zur Ablaufsteuerung**

Beschreibt den Aufbau bzw. die Darstellung und Steuerung der Transformationsprozesse mit der Benutzerschnittstelle. Beschreibung ohne Wertung.

Kategorie B.2: Expressivität der verwendeten Abbildungssprache

Bewertungskategorie B.2 beinhaltet Kriterien zur Klassifizierung der verwendeten Abbildungssprache der Software.

- **B.2.0 Gesamtwertung der Expressivität der Abbildungssprache**

Trifft eine Gesamtwertung über alle Kriterien zur Expressivität der Abbildungssprache im gewählten Transformationsfall.

0: Abbildungssprache ist nicht geeignet

1: Abbildungssprache ist geeignet

- **B.2.1 Repräsentation der Abbildungssprache**

Nennung und Angabe von Vorteilen und Nachteilen der verwendeten Abbildungssprache.

0: Abbildungssprache weist vorwiegend nachteilige Eigenschaften auf

1: Abbildungssprache weist vorwiegend vorteilhafte Eigenschaften auf

2: Abbildungssprache weist hervorragende Eigenschaften auf

- **B.2.2 Paradigma der Abbildungssprache**

Nennung und Angabe von Vorteilen und Nachteilen des Sprachparadigmas der verwendeten Abbildungssprache.

0: Sprachparadigma der Abbildungssprache wirkt sich nachteilig aus

1: Sprachparadigma der Abbildungssprache wirkt sich kaum bis nicht nachteilig aus

- **B.2.3 Handhabung anderer Sprachparadigmen**

Feststellung der Kompatibilität zu Datensätzen und Schemata abweichender Sprachparadigmen zum gewählten Transformationsfall.

0: Abbildungssprache ist nicht kompatibel

1: Abbildungssprache ist kompatibel

- **B.2.4 Versionskompatibilität zu Eingangsdaten und -schemata**

Feststellung der Kompatibilität der Abbildungssprache mit eingelesenen Datensätzen und Schemata in den Versionen des gewählten Transformationsfalls.

0: Abbildungssprache ist nicht kompatibel

1: Abbildungssprache ist kompatibel

Kategorie B.3: Transformationskompetenz der Software

Bewertungskategorie B.3 beinhaltet Kriterien zur Prüfung der Kompetenz der Software bei der Durchführung von Datenmodelltransformationen.

- **B.3.0 Gesamtwertung der Transformationskompetenz**

Trifft die Gesamtwertung aus allen Teilkriterien dieser Bewertungskategorie, um die qualifizierende Kompetenz der Software für Datenmodelltransformationen im gewählten Transformationsfall festzustellen.

0: Transformationskompetenz ist ungenügend oder mangelhaft

1: Transformationskompetenz ist ausreichend

- **B.3.1 Datenmanagement**

Beschreibt die Handhabung von Datensätzen und Schemata durch die Software beim Einlesen, zur Laufzeit und bei der Ausgabe, sowie die Möglichkeiten des Zugriffs und der Manipulation darauf.

0: Datenmanagement ist vor dem Nutzer verborgen

1: Datenmanagement ist für den Nutzer nur bedingt zugänglich

2: Datenmanagement ist für den Nutzer umfassend zugänglich

- **B.3.2 Aktualität der Funktionalität**

Prüft die Update-Mechanismen der Software.

0: Software wird nicht via Updates aktuell gehalten

1: Software wird via Updates aktuell gehalten

- **B.3.3 Erweiterbarkeit der Abbildungssprache**

Prüft die Möglichkeit der Erweiterbarkeit der Abbildungssprache mit eigenen Strukturen durch den Anwender.

0: Abbildungssprache ist nicht erweiterbar

1: Abbildungssprache ist erweiterbar

- **B.3.4 Berichtswesen**

Beschreibt das Vorhandensein und die Qualität von Dokumentationsvorgängen der Software bei der Durchführung von Datenmodelltransformationen.

0: Dokumentationsvorgänge sind nicht oder unzureichend vorhanden

1: Dokumentationsvorgänge sind ausreichend vorhanden

- **B.3.5 Qualitätsmanagement**

Beschreibt interne Mechanismen des Programms zur Validierung der erzeugten Resultate.

0: Mechanismen sind nicht oder unzureichend vorhanden

1: Mechanismen sind ausreichend vorhanden

- **B.3.6 Vorhandene Spezialisierungen für Fachanwendungen**

Beschreibt vorhandene Funktionalitäten der Software zur Anwendung für spezifische Anwendungsfälle.

0: Spezialisierung für Anwendungsfall ist nicht oder unzureichend vorhanden

1: Spezialisierung für Anwendungsfall ist vorhanden

Kategorie B.4: Funktionsbibliothek der Software

Bewertungskategorie B.4 beschreibt die in der Software enthaltene Funktionsbibliothek zur Festlegung der Abbildungen zwischen den semantischen Relationen.

- **B.4.0 Gesamtwertung der Funktionsbibliothek**

Trifft die Gesamtwertung über die Tauglichkeit der Funktionsbibliothek gemäß den Anforderungen im gewählten Transformationsfall, nach Prüfung aller Teilkriterien der Kategorie B.4.

0: Funktionsbibliothek ist ungenügend bis mangelhaft

1: Funktionsbibliothek ist ausreichend

- **B.4.1 Umfang der Funktionsbibliothek**

Prüft den Umfang der Funktionsbibliothek hinsichtlich der benötigten Konstrukte für den gewählten Transformationsfall und darüber hinaus.

0: Funktionsbibliothek beinhaltet nicht oder kaum die benötigten Konstrukte

1: Funktionsbibliothek beinhaltet die benötigten Konstrukte

2: Funktionsbibliothek beinhaltet wesentlich mehr Konstrukte als benötigt

- **B.4.2 Funktionen zur Bearbeitung der Geometrie**

Prüft die Funktionsbibliothek hinsichtlich vorhandener Konstrukte zur Bearbeitung von Geometrie.

0: Geometrie kann nicht gelesen oder verarbeitet werden

1: Geometrie kann verarbeitet werden

2: Geometrie kann erweitert werden

- **B.4.3 Funktionen zur Bearbeitung von Inhalten**

Prüft die Funktionsbibliothek hinsichtlich vorhandener Konstrukte zur Bearbeitung und zum Hinzufügen von Inhalt während der Transformation.

0: Inhalte können nicht gelesen oder verarbeitet werden

1: Inhalte können verarbeitet werden

2: Inhalte können erweitert werden

- **B.4.4 Initialisierung von Bedingungen und Schleifen**

Prüft die Funktionsbibliothek hinsichtlich dem Vorhandensein von Programmstrukturen zum Anlegen von logischen Bedingungen, Verzweigungen und Schleifen bei der Festlegung der Transformationen.

0: Bedingungen und Schleifen sind nicht oder unzureichend vorhanden

1: Bedingungen sind vorhanden

2: Bedingungen und Schleifen sind vorhanden

- **B.4.5 Qualität der Funktionsbibliothek**

Beschreibt die Qualität der Funktionsbibliothek hinsichtlich der Anwendungs- und Anpassungsmöglichkeiten der Funktionen.

0: Qualität ist ungenügend oder mangelhaft

1: Qualität ist ausreichend

2: Qualität ist hervorragend

- **B.4.6 Erweiterbarkeit mit eigenen Funktionen**

Gibt an, ob eigene Funktionen oder Skripte erstellt und angewendet werden können.

0: Funktionsbibliothek ist nicht oder unzureichend erweiterbar

1: Funktionsbibliothek ist erweiterbar

Kategorie B.5: Benutzerfreundlichkeit der Software

In dieser Bewertungskategorie wird die Benutzerfreundlichkeit der Software aus Sicht des Anwenders bewertet.

- **B.5.0 Gesamtwertung der Benutzerfreundlichkeit**

Trifft eine Gesamtwertung in der Bewertungskategorie zur Feststellung der Benutzerfreundlichkeit.

0: Benutzerfreundlichkeit ist ungenügend oder mangelhaft

1: Benutzerfreundlichkeit ist ausreichend

- **B.5.1 Form der Abbildungsfestlegung und Durchführung**

Trifft eine Aussage darüber, in welcher Form dem Anwender die Einstellungen zur Durchführung von Datenmodelltransformationen durch die Software ermöglicht wird.

0: Abbildung und Durchführung wird durch Konsolenanwendung unterstützt

1: Abbildung wird durch grafische Benutzerschnittstelle unterstützt

- **B.5.2 Mögliche Anwenderführung**

Beschreibt die Unterstützung der Software für den Anwender durch intuitive Hilfestellungen und geführte Dialogsteuerung.

0: Anwender ist völlig frei

1: Anwender wird durch Dialogsteuerung o.Ä. geführt

- **B.5.3 Grad des benötigten Know-hows zur Anwendung**

Gibt an, in welchem Grad der Anwender über Kenntnisse in der Fachanwendung respektive Programmierung verfügen muss, um Datenmodelltransformationen mit der Software umstandslos verwirklichen zu können.

- 0: Fachkenntnisse sind in Fachanwendung und Programmierung erforderlich
- 1: Fachkenntnisse sind in Fachanwendung erforderlich
- 2: Fachkenntnisse sind nicht erforderlich

- **B.5.4 Ausmaß der möglichen Automatisierung**

Beschreibt die Möglichkeiten zur Automatisierbarkeit der Transformationsvorgänge und sonstigen Arbeitsschritte.

- 0: Automatisierung ist nicht möglich
- 1: Automatisierung ist bedingt möglich
- 2: Automatisierung ist umfassend möglich

- **B.5.5 Dokumentation**

Stellt fest, ob und in welchem Umfang eine Dokumentation zur Software vorhanden ist.

- 0: Dokumentation ist nicht oder unzureichend vorhanden
- 1: Dokumentation ist ausreichend vorhanden
- 2: Dokumentation ist ausreichend vorhanden und wird außerordentlich unterstützt

- **B.5.6 Unterstützung durch Support**

Gibt an, ob ein Support durch den Entwickler/Publisher zur Beantwortung von Fragen und Lösung von Problemen angeboten wird.

- 0: Support wird nicht oder unzureichend angeboten
- 1: Support wird angeboten

- **B.5.7 Sonstige zusätzliche Funktionen**

Beschreibt weitere Features der Software, die nicht direkt für die Durchführung von Transformationen benötigt werden, aber nützlich sind.

- 0: Sonstige Funktionen sind nicht gegeben
- 1: Sonstige Funktionen sind gegeben

Kategorie B.6: Schnittstellen der Software

Die Bewertungskategorie bewertet Schnittstellen der Software zum Datentransfer nach innen und außen, sowie zur Interaktion und Kommunikation mit externen Diensten.

- **B.6.0 Gesamtwertung zu den Schnittstellen**

Trifft eine insgesamt bewertende Aussage zum Nutzen und der Tauglichkeit der Schnittstellen.

- 0: Schnittstellen sind nicht ausreichend
- 1: Schnittstellen sind ausreichend

- **B.6.1 Enthaltene Schnittstellen**

Prüft auf das Vorhandensein von Schnittstellen zum Import und Export von Datensätzen und Schemata.

0: Schnittstellen zum Import/Export sind nicht oder unzureichend vorhanden

1: Schnittstellen zum Import/Export sind vorhanden

- **B.6.2 Kompatibilität mit Web Services**

Stellt die Kompatibilität der Software respektive ihrer Schnittstellen zu externen Diensten fest.

0: Schnittstellen sind nicht kompatibel zu aktuellen Standards für Web Services

1: Schnittstellen sind kompatibel zu aktuellen Standards für Web Services

- **B.6.3 Routinen zum Einlesen von Datenformaten**

Beschreibt die Möglichkeiten Schnittstellen zum Import in die Transformationen einzubinden.

0: Routinen sind nicht oder unzureichend vorhanden

1: Routinen sind vorhanden

- **B.6.4 Routinen zum Export von Datenformaten**

Beschreibt die Möglichkeiten Schnittstellen zum Export in die Transformationen einzubinden.

0: Routinen sind nicht oder unzureichend vorhanden

1: Routinen sind vorhanden

Bewertungskatalog C: Sonstige Kategorien.

Kategorie C.1: Anforderungen durch den Anwender

In dieser Bewertungskategorie werden subjektiv durch den Anwender gestellte Anforderungen an die Software zusammengefasst.

- **C.1.0 Gesamtwertung der spezifischen Anforderungen durch den Anwender**

Die Kategorie ermöglicht die zusammenfassende Bewertung der hier festgelegten spezifischen Anforderungen.

0: Spezifische Anforderungen werden nicht oder unzureichend erfüllt

1: Spezifische Anforderungen werden erfüllt

- **C.1.1 Kompatibilität zu Betriebssystemen**

Stellt fest, ob die Software in einer zum Betriebssystem des Anwenders kompatiblen Version angeboten wird.

0: Software ist nicht oder unzureichend kompatibel zum Betriebssystem

1: Software ist kompatibel zu Betriebssystem

- **C.1.2 Spezialisierung der Software**

Gibt an, ob die Software für den Anwendungsfall spezialisiert ist oder auch für weitere Zwecke implementiert wurde.

i) spezialisiert ii) Mehrzweck

- **C.1.3 Mögliche Interaktionen mit anderer Software**

Beschreibt die Möglichkeiten zur Einbindung der Software in eigene Architekturen.

- 0: Interaktionen sind nicht oder unzureichend möglich
- 1: Interaktionen sind bedingt möglich
- 2: Interaktionen sind weitreichend möglich

- **C.1.4 Anwender-Community zum Austausch von Fachkenntnissen**

Beschreibt das Vorhandensein und die Aktivität einer Community zur Software, beispielsweise zum Austausch und gemeinschaftlichen Lösung von Problemstellungen in Internetforen.

- 0: Community ist nicht vorhanden
- 1: Community ist vorhanden
- 2: Community ist vorhanden und aktiv

Kategorie C.2: Entwicklungsstand der Software

Mit den Kriterien dieser Bewertungskategorie wird der Entwicklungsstatus hinsichtlich der Weiterentwicklung der Software festgestellt.

- **C.2.0 Gesamtwertung des Entwicklungsstandes**

Trifft eine Gesamtwertung darüber, ob der Entwicklungsstand dem Anwender genügt und ob gegebenenfalls Weiterentwicklungen gewünscht sind.

- 0: Entwicklungsstand ist nicht zufriedenstellend
- 1: Entwicklungsstand ist zufriedenstellend

- **C.2.1 Status der Projektentwicklung**

Stellt fest, ob und in welchem Umfang die Software weiterentwickelt wird, zur Anpassung an kommende Aufgabenstellungen.

- 0: Projekt wird nicht oder unzureichend weiterentwickelt
- 1: Projekt wird sukzessive weiterentwickelt
- 2: Projekt wird fortlaufend weiterentwickelt

- **C.2.2 Regelmäßigkeit der Softwareaktualisierung**

Umschreibt die Regelmäßigkeiten der Aktualisierung der Software bezüglich der Fehlerbehandlung und Weiterentwicklung.

- 0: Aktualisierung findet nicht oder unzureichend statt
- 1: Aktualisierung findet sukzessive statt
- 2: Aktualisierung findet fortlaufend statt

Kategorie C.3: Kostenstruktur der Software

In dieser Bewertungskategorie werden die Kosten beschreibend abgeschätzt, welche die Software bei der Anschaffung und im Betrieb verursacht.

- **C.3.0 Gesamtwertung der Kostenstruktur**

Die Gesamtwertung dieser Kategorien bewertet zusammenfassend die möglichen Kosten.

0: Es fallen Kosten an

1: Es fallen keine Kosten an

- **C.3.1 Anschaffungskosten der Software**

Stellt fest ob und in welchem Umfang eine Gebühr bei der Erstanschaffung der Lizenz zu entrichten ist.

0: Es fallen Anschaffungskosten an

1: Es fallen vernachlässigbar geringe Anschaffungskosten an

2: Es fallen keine Anschaffungskosten an (freie Software)

- **C.3.2 Betriebskosten der Software (Lizenzgebühren)**

Gibt an, ob und in welchem Umfang mit Gebühren für die fortlaufende Gültigkeit der Lizenz zu rechnen ist.

0: Es sind Lizenzgebühren zu entrichten

1: Es sind vernachlässigbar geringe Lizenzgebühren zu entrichten

2: Es fallen keine Lizenzgebühren an

- **C.3.3 Betriebskosten der Software (Personalstunden)**

Umschreibt den Aufwand, der durch das Personal im operationalen Betrieb der Software entsteht.

0: Personalstunden sind relevant

1: Personalstunden sind vernachlässigbar gering

2: Personalstunden sind vernachlässigbar (vollständige Automatisierung)

- **C.3.4 Lebenszyklus der Software**

Stellt fest, ob Aktualisierungen vornehmlich durch neue Versionen der Software mit eigenständiger Lizenz angeboten werden und die aktuelle Version mit ihrer Funktionalität deshalb schneller veraltet.

0: Lebenszyklus ist gering durch enge Bindung von Aktualität an neue Lizenzen

1: Lebenszyklus ist hoch durch keine oder lose Bindung von Aktualität an neue Lizenzen

6.3 Darstellung und Beschreibung des Bewertungsschemas

In diesem Abschnitt erfolgt der Entwurf eines Bewertungsschemas zur Beurteilung der verwendeten Datenmodelltransformationswerkzeuge, auf Grundlage der in den Abschnitten 6.3.1 und 6.3.2 ermittelten Kriterien. Das Schema wird in zwei Varianten festgelegt: Als tabellarische Aufzählung der Kriterien und als grafisches Verzweigungsdiagramm. Im digitalen Anhang (Anhang B) ist die vollständige Tabelle als Punkteschema (Anhang E) wiedergegeben und das Verzweigungsdiagramm (Anhang F) vollständig abgebildet.

Bei genauer Betrachtung der Kriterien wird offensichtlich, dass die Wertung nicht immer klar zu treffen ist und mitunter einer Abwägung unterliegt. Ebenso hängt die Wichtigkeit eines Kriteriums auch vom gewählten Anwendungsfall ab, da unterschiedliche Anwendungsfälle auch unterschiedliche Anforderungen an die ETL-Tools stellen. Um diesem Sachverhalten Rechnung zu tragen werden die Kriterien aus dem Katalog II (Bewertungsgruppen B und C) im Folgenden noch um einige Angaben erweitert. Dazu wird eine klassifizierende Differenzierung anhand des Typus der Kriterien vorgenommen. Es werden drei Typen unterschieden: Pflichtkriterium (PK), normales Kriterium (NK) und optionales Kriterium (OK). Ein als Pflichtkriterium klassifiziertes Kriterium ist zwingend zu erfüllen (Wertung aus den Bewertungsschlüsseln: 1 oder 2), weil der Gegenstand der Bewertung zur Lösung der Transformationsaufgabe unbedingt benötigt wird. Eine Nichterfüllung des Kriteriums führt folglich zur disqualifizierenden Bewertung des ETL-Tools. Durch ein normales Kriterium wird ausgedrückt, dass der bewertete Gegenstand nicht zwingend erforderlich ist, aber gegebenenfalls wesentlich zur Durchführung von Transformationen beiträgt. Hingegen kann der Bewertungsgegenstand eines optionalen Kriteriums je nach Anforderungen durch den Anwender angegeben oder auch vernachlässigt werden. Zur Wertung der Kriterien ist in den Katalogen zu jedem Kriterium ein Bewertungsschlüssel angegeben. Aus den Resultaten und gewonnenen Erkenntnissen zu den durchgeführten Transformationen kann eine geeignete Wahl aus den Bewertungsschlüsseln getroffen werden.

Manche Kriterien können trotz negativer Bewertung über Umwege doch noch erfüllt werden, indem andere Kriterien dazu beitragen. Beispielsweise kann das Fehlen von benötigten Funktionen zur Bewältigung einer bestimmten Aufgabe abgefangen werden, wenn die Möglichkeit besteht, eigene Funktionen oder Skripte zur Erfüllung dieser Aufgabe zu gestalten. Im grafischen Verzweigungsdiagramm wird dies durch Angabe einer Referenz auf alternative Kriterien verwirklicht. Die Referenz verweist auf ein alternativ zu erfüllendes Kriterium (Nr.) und ist hinter der Angabe zum Kriteriumstypen hinterlegt. Im konkreten Beispiel sieht die tabellarische Anordnung, mit einem Beispielkriterium aus Katalog II, wie folgt aus (Tabelle 6.2):

| B.4.3 Funktionen zur Bearbeitung von Inhalten | |
|--|---|
| Wertung: 0 | Pflichtkriterium |
| Begründung | Es fehlen Funktionen, um Texte (String) aus zwei Quellattributen zu einem Zielattribut durch Einfügen von Zeichen (Character) zu verknüpfen |
| Referenz | B.4.6 Erweiterbarkeit mit eigenen Funktionen |

Tab. 6.2 – Beispiel zur tabellarischen Darstellung für ein Bewertungskriterium

Im tabellarischen Schema erfolgt die Bewertung strikt durch Auswahl einer Wertung aus dem jeweiligen Bewertungsschlüssel der Kataloge und einer Begründung, warum die entsprechende Wertung gewählt wurde. Erst aus dem grafischen Verzweigungsdiagramm werden die Zusammenhänge zwischen einigen Kriterien deutlich.

Das Beispiel aus Tabelle 6.2 stellt sich im Verzweigungsdiagramm folgendermaßen dar (Abbildung

6.1):

| B.4.3 Funktionen zur Bearbeitung von Inhalten | | |
|---|--|--------|
| Pflichtkriterium | Faktor: 2,5 | Punkte |
| 0 | Inhalte können nicht gelesen oder verarbeitet werden | (0) |
| 1 | Inhalte können verarbeitet werden | (2,5) |
| 2 | Inhalte können erweitert werden | (5) |

| B.4.6 Erweiterbarkeit mit eigenen Funktionen | | |
|--|---|--------|
| Normales Kriterium | Faktor: 1 | Punkte |
| 0 | Funktionsbibliothek ist nicht oder unzureichend erweiterbar | (0) |
| 1 | Funktionsbibliothek ist erweiterbar | (1) |

Falls Wertung 0 bei B.4.3 muss B.4.6 mit 1 gewertet werden, ansonsten ist das Pflichtattribut nicht erfüllt

Abb. 6.1 – Beispiel zweier Elemente des grafischen Verzweigungsdiagramms

Zudem soll im Bewertungsschema die Möglichkeit gegeben werden, durch eine Punktwertung eine aussagekräftige vergleichende Bewertung mehrerer ETL-Tools vornehmen zu können. Dazu können die Zahlangaben der Bewertungsschlüssel (0 bis 2) als zu erzielende Punkte aufgefasst werden. Grundsätzlich setzen sich bei der Bewertung eines Tools alle Wertungen zu einer Gesamtpunktzahl der jeweiligen Bewertungskategorie zusammen. Die Höhe der Einzelpunktzahl kann je nach Kriterium 0 bis 2 betragen. Für Kategorie C.3 ergibt sich beispielsweise eine mögliche Maximalpunktzahl von 8 Punkten und eine mögliche Minimalpunktzahl von 0 Punkten. Weiterhin wird den Kriterien ein Faktor hinzugefügt, mit dem eine Gewichtung der Kriterien nach eigenen Vorgaben möglich ist. Der Faktor soll bei 0 beginnen (z.B. um optionale Kriterien von der Wertung auszuschließen) und in Schritten von 0,5 beliebig erhöht werden können. Die Gewichtung bleibt dem Anwender überlassen. Zwar können sich bedingt durch die Faktoren unterschiedliche Gesamtpunktzahlen ergeben, aber ein festgelegtes Bewertungsschema lässt sich auf mehrere Tools anwenden, so dass deren erreichte Gesamtpunktzahl vergleichbar ist. Im obigen Beispiel aus Abbildung 6.2 würde aus beiden dargestellten Kriterien mit den angegebenen Faktoren eine maximale Punktzahl von 6 Punkten resultieren. Würde B.4.3 mit Wertung 0 bedacht werden, ergäbe sich aus dem dann alternativ herangezogenen Kriterium B.4.6 eine maximale Wertung von 1 Punkt.

Das grafische Bewertungsschema wurde beispielhaft mit der Bewertung von FME ausgefüllt im digitalen Anhang hinterlegt (siehe Anhang B).

6.4 Bewertung von Feature Manipulation Engine

In den nachfolgenden Abschnitten wird FME nach den entwickelten Katalogen evaluiert und bewertet. Es wird zu jedem identifizierten Kriterium die Wertung angegeben und eine Begründung angeführt,

weshalb die Wertung gewählt wurde. Die Bewertung bezieht sich auf alle drei Transformationsfälle zusammen.

6.4.1 Bewertung von FME nach Bewertungskatalog A

| Katalog A: Kriterien zur Evaluierung der Ergebnisse | |
|---|--|
| A.1 Gesamtwertung der Evaluierung der Ergebnisse | |
| 1: Evaluierung stellt korrekte Resultate fest | |
| Begründung | Die Ergebnisse sind valide und wohlgeformt und die Objektorientierung wird in den Zieldaten korrekt hergestellt. |
| A.1 Prüfung der Wohlgeformtheit zum Zielmodell | |
| 1: Daten sind wohlgeformt | |
| Begründung | Sowohl die interne Prüfung mit dem <i>XMLValidator</i> , als auch externe Überprüfungen mit XML-Editoren konnten eine Wohlgeformtheit feststellen. |
| A.2 Prüfung auf Validierung zum Zielmodell | |
| 1: Daten sind valide | |
| Begründung | Die Daten sind valide. |
| A.3 Prüfung auf semantische Vollständigkeit | |
| 1: Daten sind semantisch vollständig | |
| Begründung | Es konnten alle Inhalte erzeugt werden. |
| A.4 Prüfung der geometrischen Transformation | |
| 1: Geometrie ist korrekt | |
| Begründung | Die Geometrie wurde korrekt erzeugt, war in den beiden Transformationsfällen allerdings auch unproblematisch. Bei komplexeren originären Geometrien könnte es zu Fehlern kommen (vgl. [Popp, 2015]). |
| A.5 Prüfung der Koordinatenreferenz-Transformation | |
| 1: Koordinatenreferenz ist korrekt | |
| Begründung | Die Reprojektierung der Koordinatenreferenzsysteme wurde strichprobenartig als korrekt attestiert. |
| A.6 Prüfung der Identifikator-Zuweisung | |
| 1: Objektidentifikatoren sind korrekt | |
| Begründung | Die <code>gml:ids</code> werden nach korrekter Kodierung erzeugt bzw. aus den originären Datensätzen übernommen. Eine weitere Erzeugung von Identifikatoren war nicht notwendig. |
| A.7 Prüfung der Kodierung | |
| 1: Kodierung ist korrekt | |
| Begründung | Alle durch FME intern abgebildeten Attributwerte wurden anschließend in den Zieldaten korrekt kodiert. Vom Anwender erzeugte Attributwerte (z.B. String) müssen vom Anwender selbst korrekt kodiert werden (z.B. utf-8). |

| A.8 Prüfung der Objekthierarchien | |
|--|--|
| 1: Objektorientierung ist korrekt | |
| Begründung | Trotz interner relationaler Strukturierung in FME konnten die Zieldaten wieder objektorientiert nach den Vorgaben des Zielschemas angeordnet werden. |

Tab. 6.3 – FME: Bewertung der Software nach Bewertungskatalog A: Evaluierung der Ergebnisse

6.4.2 Bewertung von FME nach Bewertungskatalog B

| Katalog B: Leistungsvermögen der Software | |
|---|--|
| B.1 Transformationsansatz der Software | |
| B.1.0 Gesamtwertung des Transformationsansatzes | |
| 1: Transformationsansatz ist ausreichend | |
| Begründung | Nach Betrachtung aller Wertungen kann ein ausreichender Transformationsansatz attestiert werden. |
| B.1.1 Systemabhängigkeit | |
| 0: Systemabhängigkeit ist gegeben | |
| Begründung | FME ist eine proprietäre Software, die zudem intern ein eigenes Datenmodell verwendet. |
| B.1.2 Mögliche Abbildungsebenen | |
| 2: Transfermatebene (Abbildung) und Datenebene (Anwendung) | |
| Begründung | Die Transformationsregeln werden zwischen den Anwendungsschemata (XSD) definiert und anschließend auf eingelesene Datensätze zur Transformation angewandt. |
| B.1.3 Initialisierbare Richtungen der Transformation | |
| i: undirektional; iii: horizontal | |
| Begründung | Die Transformationen werden fix zwischen Quell- und Ziel-Anwendungsschema der gleichen Ebene definiert. |
| B.1.4 Arten der möglichen Transformationsausführung | |
| i: offline | |
| Begründung | Als Desktop-Anwendung führt FME die Transformationen offline aus. |
| B.1.5 Skalierbarkeit der Transformation | |
| Begründung | Die Skalierbarkeit ist vor allem durch Beeinflussung der Größe des Quelldatensatzes möglich. Je größer der Datensatz, desto beanspruchender ist die Transformation. |
| B.1.6 Mögliche Strukturen zur Ablaufsteuerung | |
| Begründung | Es sind ausreichend Möglichkeiten vorhanden, um Transformationsvorgänge zu steuern. Die Software stellt eine GUI und zahlreiche Einstellungsmöglichkeiten zur Verfügung. |

| | |
|---|---|
| B.2 Expressivität der verwendeten Abbildungssprache | |
| B.2.0 Gesamtwertung der Expressivität der Abbildungssprache | |
| 1: Abbildungssprache ist geeignet | |
| Begründung | Die Abbildungen werden durch XMap vorgenommen und funktionierten in den ausgewählten Anwendungsfällen tadellos. Das interne Schema von FME funktioniert ebenfalls korrekt, jedoch ist die Verarbeitung von Geometrien zu prüfen (vgl. [Popp 2015]). |
| B.2.1 Repräsentation der Abbildungssprache | |
| 1: Abbildungssprache weist vorwiegend vorteilhafte Eigenschaften auf | |
| Begründung | XMap funktionierte im praktischen Teil dieser Arbeit weitestgehend einwandfrei. Bei der Verarbeitung von Geometrien scheinen jedoch noch Schwächen vorzuliegen. |
| B.2.2 Paradigma der Abbildungssprache | |
| 1: Sprachparadigma der Abbildungssprache wirkt sich kaum bis nicht nachteilig aus | |
| Begründung | XMap ist eine Skriptsprache aus dem XML-Paradigma und scheint für das Transferformat GML gut geeignet. |
| B.2.3 Handhabung anderer Sprachparadigmen | |
| 1: Abbildungssprache ist kompatibel | |
| Begründung | Die Schemata und Datensätze werden unabhängig vom Sprachparadigma in ein internes relationales Schema übertragen. Der Paradigmenwechsel erfolgte einwandfrei. |
| B.2.4 Versionskompatibilität zu Eingangsdaten und -schemata | |
| 1: Abbildungssprache ist kompatibel | |
| Begründung | Alle Datensätze und Schemata konnten in den vorliegenden Versionen sowohl gelesen als auch geschrieben werden. |
| B.3 Transformationskompetenz der Software | |
| B.3.0 Gesamtwertung der Transformationskompetenz der Software | |
| 1: Transformationskompetenz ist ausreichend | |
| Begründung | FME lieferte die besten Ergebnisse aus den Transformationsfällen, auch unter Berücksichtigung der Ergebnisse aus Popp (2015). |
| B.3.1 Datenmanagement | |
| 2: Datenmanagement ist für den Nutzer umfassend zugänglich | |
| Begründung | Das Datenmanagement ist variabel und vom Nutzer umfassend beeinflussbar, auch deshalb, da die zur Laufzeit erzeugten Dokumente einsehbar sind. |
| B.3.2 Aktualität der Funktionalität | |
| 1: Software wird via Updates aktuell gehalten | |
| Begründung | FME erscheint regelmäßig in einer neuen Edition, die nicht neu lizenziert ist. Zudem werden in die Editionen Updates eingepflegt. |
| B.3.3 Erweiterbarkeit der Abbildungssprache | |

| | |
|---|--|
| 1: Abbildungssprache ist erweiterbar | |
| Begründung | Es besteht die Möglichkeit auch eigene XFMap oder XML-Path Dokumente einzubinden. Inwiefern diese an die internen Datenmodelle von FME gebunden sind ist unklar. |
| B.3.4 Berichtswesen | |
| 1: Dokumentationsvorgänge sind ausreichend vorhanden | |
| Begründung | Alle Vorgänge werden während der Transformation verständlich dokumentiert. Fehlermeldungen sind ebenfalls so ausformuliert, dass die Fehlerquelle identifiziert werden kann. |
| B.3.5 Qualitätsmanagement | |
| 1: Mechanismen sind ausreichend vorhanden | |
| Begründung | Es steht eine Funktion zur Verfügung (<i>XMLValidator</i> , um die resultierenden Datensätze zu validieren. Falls die Validierung scheitert werden dabei auch die Gründe angegeben. Auch im Report zur Transformation sind detaillierte Informationen enthalten. |
| B.3.6 Vorhandene Spezialisierungen für Fachanwendungen | |
| 1: Spezialisierungen für Anwendungsfälle sind vorhanden | |
| Begründung | Die Zielschemata zu den Anwendungsfällen sind auch intern in FME hinterlegt. Insbesondere steht INSPIRE GML als Output-Format zur Verfügung. |
| B.4 Funktionsbibliothek der Software | |
| B.4.0 Gesamtwertung der Funktionsbibliothek | |
| 1: Funktionsbibliothek ist ausreichend | |
| Begründung | Mit mehr als 400 Funktionen fällt die Bibliothek von FME enorm umfangreich aus. Die Qualität der Funktionen war in jedem Fall ausreichend. Zudem sind die Funktionen durch mögliche Einstellungen sehr variabel und anpassbar. Es sind auch viele Programm-Features bereits in die Funktionen integriert (z.B. Texteditor zur Erzeugung von Werten). |
| B.4.1 Umfang der Funktionsbibliothek | |
| 2: Funktionsbibliothek beinhaltet wesentlich mehr Konstrukte als benötigt | |
| Begründung | FME bietet sehr viel mehr Möglichkeiten als HALE bzw. GeoKettle und sehr viel mehr Funktionen als benötigt wurden. |
| B.4.2 Funktionen zur Bearbeitung der Geometrie | |
| 2: Geometrie kann verarbeitet und erweitert werden | |
| Begründung | Allein in der Rubrik Geometrie sind 30 Funktionen enthalten. Es können auch geometrische Operationen vorgenommen und neue Geometrien erstellt werden. |
| B.4.3 Funktionen zur Bearbeitung von Inhalten | |
| 2: Inhalte können erweitert werden | |

| | |
|---|---|
| Begründung | Die Möglichkeiten zur Bearbeitung und Erzeugung von Inhalten sind umfangreich und vielfältig, jedoch an das Zielschema gebunden. Mit <i>XQuery</i> können zudem weitere Inhalte eingefügt werden (was allerdings für «CI_Citation» nicht funktionierte). Es ist zu prüfen, ob FME mit Schemaerweiterungen umgehen kann. |
| B.4.4 Initialisierung von Bedingungen und Schleifen | |
| 2: Bedingungen und Schleifen sind vorhanden | |
| Begründung | Es können in Form von Funktionen Bedingungen und Schleifen eingebunden werden. Auch in anderen Funktionen sind solche Konstrukte teilweise enthalten, beispielsweise zur bedingten Belegung von Attributwerten. |
| B.4.5 Qualität der Funktionsbibliothek | |
| 2: Qualität ist hervorragend | |
| Begründung | Die Funktionen funktionieren gut, sind ausreichend dokumentiert und ohne Aufwand oftmals intuitiv bedien- und arrangierbar. Für komplexere Funktionen, wie z.B. <i>XMLQuery</i> , ist jedoch Fachkenntnis nötig. |
| B.4.6 Erweiterbarkeit mit eigenen Funktionen | |
| 1: Funktionsbibliothek ist erweiterbar | |
| Begründung | Es können mit Python oder Java eigene neue Funktionen in die Bibliothek integriert werden. Außerdem ist es möglich die Einstellungen zu den Funktionen zu personalisieren und dann wieder zu verwenden. |
| B.5 Benutzerfreundlichkeit der Software | |
| B.5.0 Gesamtwertung der Benutzerfreundlichkeit | |
| 1: Benutzerfreundlichkeit ist ausreichend | |
| Begründung | Die Benutzerfreundlichkeit wird durch FME ausreichend gewährleistet. Bei Problemstellungen hilft oftmals die Dokumentation weiter, falls nicht, steht ein Support durch den Anbieter zur Verfügung. Die Bedienung der Software ist schnell erlernbar. Für die Durchführung von Datenmodelltransformationen werden unbedingt Fachkenntnisse benötigt. |
| B.5.1 Form der Abbildungsfestlegung und Durchführung | |
| 1: Abbildung wird durch grafische Benutzerschnittstelle unterstützt | |
| Begründung | FME stellt eine sehr gut funktionierende GUI mit vielen Extras (z.B. Kommentarfunktion, Gruppierung) zur Verfügung. Alle Funktionen sind durch grafische Repräsentationen in Form von Boxen (u.s. <i>Transformern</i> , <i>Reader</i> , <i>Writer</i>) frei auf der Arbeitsfläche arrangierbar. Die Ablaufsteuerung wird intuitiv durch Verbindung der Boxen ermöglicht. |
| B.5.2 Mögliche Anwenderführung | |
| 0: Anwender ist völlig frei | |
| Begründung | Da FME eine Software mit weitreichenden Möglichkeiten ist, wird auch dem Anwender die Festlegung von Vorgängen frei überlassen. Allerdings ist die Bedienung der Software dennoch schnell erlernbar. |

| | |
|---|---|
| B.5.3 Grad des benötigten Know-hows zur Anwendung | |
| 1: Fachkenntnisse sind in Fachanwendung erforderlich | |
| Begründung | Fachkenntnisse im Umgang mit Transferformaten, Datenformaten etc. sollten grundlegend vorhanden sein. Darüber hinaus sind Programmierkenntnisse hilfreich, aber für einfache Fälle nicht erforderlich. Die reine Bedienung der Software ist auch für geübtes Personal ohne Fachkenntnisse in der Geoinformatik nach Einarbeitung möglich. |
| B.5.4 Ausmaß der möglichen Automatisierung | |
| 2: Automatisierung ist umfassend möglich | |
| Begründung | Es ist möglich mehrere Workspaces zu bündeln, dazu Einstellungen vorzunehmen (z.B. Zielpfade) und gemeinsam zu prozessieren. FME ist außerdem in der Lage Dateipfade automatisch auszulesen und zu erzeugen. Mit Phyton kann ein Workspace zudem extern ausgeführt werden. |
| B.5.5 Dokumentation | |
| 2: Dokumentation ist ausreichend vorhanden und wird außerordentlich unterstützt | |
| Begründung | Die Dokumentation der Software ist detailliert, reichhaltig, verständlich und wird durch zahlreiche Beispiele unterstützt. |
| B.5.6 Unterstützung durch Support | |
| 1: Support wird angeboten | |
| Begründung | Der Support kann per E-Mail und telefonisch kontaktiert werden. Es stehen Sachbearbeiter zum persönlichen Kontakt bereit. |
| B.5.7 Sonstige zusätzliche Funktionen | |
| 1: Sonstige Funktionen sind gegeben | |
| Begründung | In der verwendeten Edition ist auch der FME <i>Data Inspector</i> enthalten, mit dem man Datensätze 3-dimensional visualisieren kann. FME ist auch abseits von Datenmodelltransformationen vielfältig einsetzbar. |
| B.6 Schnittstellen der Software | |
| 1: Schnittstellen sind ausreichend | |
| B.6.0 Gesamtwertung zu den Schnittstellen | |
| 1: Schnittstellen sind ausreichend | |
| Begründung | Es sind Schnittstellen eingebunden, durch die eine Kommunikation mit Web Services und Skripten (Python) möglich ist. Es können auch Dienste wie z.B. WFS aufgerufen und Daten von Servern abgerufen bzw. auf Servern hinterlegt werden. |
| B.6.1 Enthaltene Schnittstellen | |
| 1: Schnittstellen zum Import/Export sind vorhanden | |
| Begründung | Die Reader und Writer sind in der Lage mit Web Services und Servern zu kommunizieren. |
| B.6.2 Kompatibilität mit Web Services | |
| 1: Schnittstellen sind kompatibel zu aktuellen Standards für Web Services | |

| | |
|--|---|
| Begründung | Die Reader können mit Web Services (z.B. WFS) nach aktuellen Standards in Aktion treten. |
| B.6.3 Routinen zum Einlesen von Datenformaten | |
| 1: Routinen sind vorhanden | |
| Begründung | Die Routine (Reader) ist kompakt gehalten und für die Anwendung intuitiv gestaltet. Es können Netzdienste (z.B. WFS) eingebunden werden. Zudem ist sie sehr interoperabel, da eine Vielzahl plattformunabhängiger und auch proprietärer Formate (z.B. ESRI, MS Excel) unterstützt wird. |
| B.6.4 Routinen zum Export von Datenformaten | |
| 1: Routinen sind vorhanden | |
| Begründung | Die Routine (Writer) ist kompakt gehalten und für die Anwendung intuitiv gestaltet. Zudem ist sie sehr interoperabel, da eine Vielzahl plattformunabhängiger und auch proprietärer Formate (z.B. ESRI, MS Excel) unterstützt wird. |

Tab. 6.4 – FME: Bewertung der Software nach Bewertungskatalog B: Leistungsvermögen der Software

6.4.3 Bewertung von FME nach Bewertungskatalog C

| | |
|--|--|
| Katalog C: | Sonstige Kategorien |
| C.1 Anforderungen durch den Anwender | |
| C.1.0 Gesamtwertung der spezifischen Anforderungen durch den Anwender | |
| 1: Anforderungen werden erfüllt | |
| Begründung | Die Software ist für die gängigsten Betriebssysteme ausgelegt und für viele Einsatzzwecke anwendbar. Zudem kann die Software mit Python in Arbeitsschritte anderer Tools eingebunden werden. |
| C.1.1 Kompatibilität zu Betriebssystemen | |
| 1: Software ist kompatibel zu Betriebssystem | |
| Begründung | Die Software kann sowohl unter Windows, Linux, als auch einem Mac installiert werden. |
| C.1.2 Spezialisierung der Software | |
| ii: Mehrzweck | |
| Begründung | Die Software ist für vielfältige Zwecke in der Geoinformatik ausgelegt. |
| C.1.3 Mögliche Interaktion mit anderer Software | |
| 1: Interaktionen sind weitreichend möglich | |
| Begründung | Mit Python kann FME auch in anderer Software angesteuert werden. Die verarbeitbaren und erzeugbaren Formate sind vielfältig und auf dem aktuellen Stand. |
| C.1.4 Anwender-Community zum Austausch von Fachkenntnissen | |
| 2: Community ist vorhanden und aktiv | |
| Begründung | Es werden offizielle Foren zum Austausch angeboten. Entwickler von FME sind in den Foren ebenfalls aktiv. |

| | |
|--|--|
| C.2 Entwicklungsstand der Software | |
| C.2.0 Gesamtwertung des Entwicklungsstandes | |
| 1: Entwicklungsstand ist zufriedenstellend | |
| Begründung | Die Software wird fortlaufend weiterentwickelt und durch Updates aktuell gehalten. Die Entwickler fassen Hinweise zu Fehlern auf und beheben diese (vgl. [Popp 2015]). |
| C.2.1 Status der Projektentwicklung | |
| 2: Projekt wird fortlaufend weiterentwickelt | |
| Begründung | Es erscheinen regelmäßig Updates und einmal jährlich eine neue Edition der Software, mit denen auch neue Funktionalitäten eingebunden werden. |
| C.2.2 Regelmäßigkeit der Softwareaktualisierung | |
| 2: Aktualisierung findet regelmäßig statt | |
| Begründung | Neue Editionen erscheinen jährlich und Updates werden regelmäßig bereitgestellt, wenn der Anbieter Bedarf hierfür sieht. |
| C.3 Kostenstruktur der Software | |
| C.3.0 Gesamtwertung der Kostenstruktur | |
| 0: Es fallen Kosten an | |
| Begründung | |
| C.3.1 Anschaffungskosten der Software | |
| 2: Es fallen keine Anschaffungskosten an | |
| Begründung | Die Kosten sind im Einzelfall beim Anbieter zu erfragen und als fortlaufende Lizenzgebühr zu entrichten. Ob die Preise verhandelbar sind ist unklar. |
| C.3.2 Betriebskosten der Software (Lizenzgebühren) | |
| 0: Es fallen Lizenzgebühren an | |
| Begründung | Es fallen jährlich zu entrichtende Lizenzgebühren an. Die Auskunft muss beim Anbieter erfragt werden. |
| C.3.3 Betriebskosten der Software (Personalstunden) | |
| 1: Personalstunden sind vernachlässigbar gering | |
| Begründung | Nach Initialisierung der Abbildungen müssen diese nur nach Änderungen in den Anwendungsschemata aktualisiert werden. Ansonsten ist die Software gut automatisierbar und die Personalstunden lassen sich dadurch beschränken. |
| C.3.4 Lebenszyklus der Software | |
| 1: Lebenszyklus ist hoch durch keine Bindung von Aktualität an neue Lizenzen | |
| Begründung | Neuerungen werden meist in neue Editionen mit eigenständigen Lizenzen eingebunden und werden automatisch für Lizenzinhaber bereitgestellt. |

Tab. 6.5 – FME: Bewertung der Software nach Bewertungskatalog C: Sonstige Kategorien

6.4.4 Abschließende Bewertung von FME

FME ist als Software für Datenmodelltransformationen sehr gut geeignet. Die Einarbeitung in FME ist gut möglich und die Dokumentation ist außerordentlich gut. Mit FME kann auch ein weitaus größeres Aufgabenfeld in Betracht gezogen werden für verschiedene Aufgaben in einer Geodateninfrastruktur. Die Software stellt zudem zahlreiche Funktionalitäten zur Verfügung, welche die Arbeitsschritte erleichtern. Auch ist eine weitreichende Automatisierung von Datenmodelltransformationen möglich. FME wäre insbesondere auch dann in Betracht zu ziehen, wenn die Software für weitere Anwendungszwecke herangezogen wird, wodurch sich auch die Kosten betriebswirtschaftlich auf weitere Posten umverteilen lassen.

6.5 Bewertung von GeoKettle

In den nachfolgenden Abschnitten wird GeoKettle aufgrund der gewonnenen Erkenntnisse nach den entwickelten Katalogen evaluiert und bewertet. Es wird zu jedem identifizierten Kriterium die Wertung angegeben und eine Begründung angeführt, weshalb die Wertung gewählt wurde. Die Bewertung bezieht sich auf alle drei Transformationsfälle zusammen.

6.5.1 Bewertung von GeoKettle nach Bewertungskatalog A

| Katalog A: Kriterien zur Evaluierung der Ergebnisse | |
|---|---|
| A.0 Gesamtwertung der Evaluierung der Ergebnisse | |
| 0: Evaluierung stellt bedingt durch Fehler mangelhafte Resultate fest | |
| Begründung | Es konnten keine validen Resultate erzielt werden. |
| A.1 Prüfung der Wohlgeformtheit zum Zielmodell | |
| 1: Daten sind wohlgeformt | |
| Begründung | Die erzeugten Daten waren wohlgeformt. |
| A.2 Prüfung auf Validierung zum Zielmodell | |
| 0: Daten sind nicht valide | |
| Begründung | Die interne Prüfung mit dem <i>XMLValidator</i> ergab eine nicht näher spezifizierbare Fehlermeldung. Externe Prüfungen waren nicht valide. |
| A.3 Prüfung auf semantische Vollständigkeit | |
| 0: Daten sind semantisch nicht vollständig | |
| Begründung | Komplexe Inhalte wie z.B. «GeographicalName» können nicht gebildet werden. |
| A.4 Prüfung der geometrischen Transformation | |
| 0: Geometrie ist nicht korrekt | |
| Begründung | Die Geometrie von CityGML konnte nicht gelesen werden. Die Kodierung ist nur in GML 3.1.1 möglich. |

| | |
|---|--|
| A.5 Prüfung der Koordinatenreferenz-Transformation | |
| 0: Koordinatenreferenz ist nicht korrekt | |
| Begründung | Die Reprojektierung der Koordinatenreferenzsysteme lieferte falsche Ergebnisse. |
| A.6 Prüfung der Identifikator-Zuweisung | |
| 0: Objektidentifikatoren sind nicht korrekt | |
| Begründung | Die <code>gml:ids</code> wurden entfernt. |
| A.7 Prüfung der Kodierung | |
| 0: Kodierung ist nicht korrekt | |
| Begründung | Die Kodierung ist nicht nach aktuellen Standards möglich. |
| A.8 Prüfung der Objekthierarchien | |
| 1: Objektorientierung ist korrekt | |
| Begründung | Trotz interner relationaler Strukturierung in GeoKettle konnten die Zieldaten wieder objektorientiert angeordnet werden. Für komplexere Fälle konnte das nicht überprüft werden. |

Tab. 6.6 – GeoKettle: Bewertung der Software nach Bewertungskatalog A: Evaluierung der Ergebnisse

6.5.2 Bewertung von GeoKettle nach Bewertungskatalog B

| | |
|---|---|
| Katalog B: | Leistungsvermögen der Software |
| B.1 Transformationsansatz der Software | |
| B.1.0 Gesamtwertung des Transformationsansatzes | |
| 0: Transformationsansatz ist mangelhaft | |
| Begründung | Die Funktionen sind in der Lage mit aktuellen Standards umzugehen. Der Ansatz ist nur bedingt für Datenmodelltransformationen ausgelegt. Anwendungsschemata können nicht herangezogen werden. |
| B.1.1 Systemabhängigkeit | |
| 1: Systemabhängigkeit ist nicht gegeben | |
| Begründung | GeoKettle ist eine Open-Source-Software und frei verwendbar. |
| B.1.2 Mögliche Abbildungsebenen | |
| 0: Datenebene (Abbildung und Anwendung) | |
| Begründung | Es können keine Abbildungsregeln im eigentlichen Sinn gebildet werden. Die Transformation kann lediglich durch stückweises bearbeiten der Attributwerte und der Hierarchie ermöglicht werden. Es können Formate wie CSV gelesen werden, diese Möglichkeiten wurden im Rahmen der vorliegenden Arbeit aber nicht untersucht. |
| B.1.3 Initialisierbare Richtungen der Transformation | |
| i: undirektional; iii: horizontal | |
| Begründung | Die Transformationen werden fix zwischen Quell- und Zieldaten der gleichen Ebene definiert. |

| | |
|--|--|
| B.1.4 Arten der möglichen Transformationsausführung | |
| i: offline | |
| Begründung | Als Desktop-Anwendung führt GeoKettle die Transformationen offline aus. |
| B.1.5 Skalierbarkeit der Transformation | |
| Begründung | Die Skalierbarkeit ist nur durch Beeinflussung der Größe des Quelldatensatzes möglich. Je größer der Datensatz, desto beanspruchender ist die Transformation. Durch die Vielzahl an benötigten Funktionen zur Verwirklichung einer Transformation wird die Performance zusätzlich behindert. |
| B.1.6 Mögliche Strukturen zur Ablaufsteuerung | |
| Begründung | Es können nur unzureichend Einstellungen getroffen werden. Die Software stellt eine GUI zur Verfügung, die an FME angelehnt ist. |
| B.2 Expressivität der verwendeten Abbildungssprache | |
| B.2.0 Gesamtwertung der Expressivität der Abbildungssprache | |
| 0: Abbildungssprache ist nicht vorhanden | |
| Begründung | GeoKettle verwendet keine Abbildungssprache. |
| B.3 Transformationskompetenz der Software | |
| B.3.0 Gesamtwertung der Transformationskompetenz der Software | |
| 0: Transformationskompetenz ist mangelhaft | |
| Begründung | GeoKettle weist im Hinblick auf Datenmodelltransformationen nach aktuellen Standards nur eine mangelhafte Fähigkeit auf. |
| B.3.1 Datenmanagement | |
| 2: Datenmanagement ist für den Nutzer umfassend zugänglich | |
| Begründung | Die Datensätze können in den Tabellen vollständig eingesehen und bearbeitet werden. Zudem sind in einigen Funktionen Voransichten zu den daraus erzielten Resultaten enthalten. |
| B.3.2 Aktualität der Funktionalität | |
| 0: Software wird derzeit nicht via Updates aktuell gehalten | |
| Begründung | Seit 2,5 Jahren ist kein Update mehr erschienen. |
| B.3.3 Erweiterbarkeit der Abbildungssprache | |
| 0: Abbildungssprache ist nicht erweiterbar | |
| Begründung | Es wird keine Abbildungssprache verwendet. |
| B.3.4 Berichtswesen | |
| 0: Dokumentationsvorgänge sind unzureichend vorhanden | |
| Begründung | Es wird ein Report zum Transformationsvorgang erzeugt, der jedoch nur bedingt Erkenntnisse liefert. Fehlermeldungen sind nicht ausreichend spezifiziert, um als Anwender die Fehlerursache begründen zu können. |
| B.3.5 Qualitätsmanagement | |
| 1: Mechanismen sind unzureichend vorhanden | |

| | |
|--|--|
| Begründung | Es ist ein Validator für XML-Dokumente enthalten. Das Ergebnis der Validierung fiel jedoch immer negativ aus, auch für unbearbeitete Quelldaten. |
| B.3.6 Vorhandene Spezialisierungen für Fachanwendungen | |
| 0: Spezialisierungen für Anwendungsfälle sind nicht vorhanden | |
| Begründung | GeoKettle bietet neben den Funktionen keine weiteren Inhalte an. |
| B.4 Funktionsbibliothek der Software | |
| B.4.0 Gesamtwertung der Funktionsbibliothek | |
| 0: Funktionsbibliothek ist mangelhaft | |
| Begründung | Es fehlen Funktionen zur Bearbeitung von Geometrien und die vorhandenen Funktionen sind nur umständlich anwendbar oder funktionieren nicht bei der Ausführung mit den vorliegenden Anwendungsfällen. |
| B.4.1 Umfang der Funktionsbibliothek | |
| 0: Funktionsbibliothek beinhaltet kaum die benötigten Konstrukte | |
| Begründung | Obwohl zahlreiche Funktionen in der Bibliothek enthalten sind, fehlen doch klare Konstrukte, um Transformationen leichter durchführen zu können. |
| B.4.2 Funktionen zur Bearbeitung der Geometrie | |
| 0: Geometrie kann nicht verarbeitet werden | |
| Begründung | Es fehlen entsprechende Funktionen. |
| B.4.3 Funktionen zur Bearbeitung von Inhalten | |
| 0: Inhalte können nicht gelesen werden | |
| Begründung | GML 3.2.1 kann nicht gelesen werden. Die Erzeugung von Attributwerten ist nur eingeschränkt möglich. |
| B.4.4 Initialisierung von Bedingungen und Schleifen | |
| 2: Bedingungen und Schleifen sind vorhanden | |
| Begründung | Es sind Funktionen zur bedingten Ablaufsteuerung mit Bedingungen und Schleifen vorhanden. |
| B.4.5 Qualität der Funktionsbibliothek | |
| 0: Qualität ist mangelhaft | |
| Begründung | Die Funktionen sind darauf ausgelegt Tabellen zu manipulieren und deshalb nur unzureichend für umfangreichere Datenmodelltransformationen. |
| B.4.6 Erweiterbarkeit mit eigenen Funktionen | |
| 0: Funktionsbibliothek ist nicht erweiterbar | |
| Begründung | Es können keine eigenen neuen Funktionen in die Bibliothek integriert werden. |
| B.5 Benutzerfreundlichkeit der Software | |
| B.5.0 Gesamtwertung der Benutzerfreundlichkeit | |
| 0: Benutzerfreundlichkeit ist mangelhaft | |
| Begründung | Wegen schlechter oder fehlender Dokumentation, umständlichen Funktionen und kryptischen Fehlermeldungen ist die Benutzerfreundlichkeit kaum gegeben. |

| | |
|---|---|
| B.5.1 Form der Abbildungsfestlegung und Durchführung | |
| 1: Abbildung wird durch grafische Benutzerschnittstelle unterstützt | |
| Begründung | GeoKettle beinhaltet eine GUI, die in ihrer Gestaltung an FME erinnert. Die Anordnung von Funktionen geht darin äquivalent von staten und ist übersichtlich. |
| B.5.2 Mögliche Anwenderführung | |
| 0: Anwender ist völlig frei | |
| Begründung | Der Anwender wird bei der Durchführung von Transformationen nicht weiter durch das Programm unterstützt. |
| B.5.3 Grad des benötigten Know-hows zur Anwendung | |
| 1: Fachkenntnisse sind in Fachanwendung erforderlich | |
| Begründung | Fachkenntnisse im Umgang mit Transferformaten, Datenformaten etc. sollten grundlegend vorhanden sein. Programmierkenntnisse sind für die Einbindung von Skripten (XPath) erforderlich. Die reine Bedienung der Software ist auch für geübtes Personal ohne Fachkenntnisse in der Geoinformatik nach Einarbeitung möglich. |
| B.5.4 Ausmaß der möglichen Automatisierung | |
| 1: Automatisierung ist bedingt möglich | |
| Begründung | Eine Automatisierung wird bedingt dadurch unterstützt, dass Workflows gebündelt und gemeinsam ausgeführt werden können. |
| B.5.5 Dokumentation | |
| 1: Dokumentation ist unzureichend vorhanden | |
| Begründung | Es wird ein Report über den Transformationsvorgang erstellt. Dieser liefert allerdings nur unzureichende Informationen und bietet keine ausreichende Hilfestellung zur Identifikation von Fehlerquellen. |
| B.5.6 Unterstützung durch Support | |
| 0: Support wird nicht angeboten | |
| Begründung | Wird nicht angeboten. |
| B.5.7 Sonstige zusätzliche Funktionen | |
| 0: Sonstige Funktionen sind nicht gegeben | |
| Begründung | GeoKettle bietet keine weiteren Software-Features an. |
| B.6 Schnittstellen der Software | |
| B.6.0 Gesamtwertung zu den Schnittstellen | |
| 0: Schnittstellen sind nicht ausreichend | |
| Begründung | Es sind Schnittstellen zur Einbindung von Daten aus Datenbanken vorhanden. Das Input mit der Schnittstelle zur Interaktion mit dem WFS des LfU funktionierte in den Tests nicht. |
| B.6.1 Enthaltene Schnittstellen | |
| 1: Schnittstellen zum Import/Export sind vorhanden | |

| | |
|---|---|
| Begründung | Es können zahlreiche Datenformate gelesen und exportiert werden. Jedoch teilweise nicht nach den aktuellen Standards. Außerdem funktionierte der Datenimport nur bedingt, auch für GML 3.1.1. |
| B.6.2 Kompatibilität mit Web Services | |
| 0: Schnittstellen sind nicht kompatibel zu aktuellen Standards für Web Services | |
| Begründung | Möglicherweise ist die Funktion <i>WFS Input</i> veraltet. |
| B.6.3 Routinen zum Einlesen von Datenformaten | |
| 1: Routinen sind vorhanden | |
| Begründung | Es konnte im Transformationsfall I das Transferformat GML 3.1.1 gelesen werden. Ansonsten ist der Datenimport meist misslungen. Eine zweite Routine zum Import kann GML 3.1.1 nur bedingt lesen und wiedergeben. Darüber hinaus können viele andere Formate (z.B. Shapefile) eingelesen werden. |
| B.6.4 Routinen zum Export von Datenformaten | |
| 1: Routinen sind vorhanden | |
| Begründung | Es kann im Format GML 3.1.1 exportiert werden. |

Tab. 6.7 – GeoKettle: Bewertung der Software nach Bewertungskatalog B: Leistungsvermögen der Software

6.5.3 Bewertung von GeoKettle nach Bewertungskatalog C

| | |
|--|--|
| Katalog C: | Sonstige Kategorien |
| C.1 Anforderungen durch den Anwender | |
| C.1.0 Gesamtwertung der spezifischen Anforderungen durch den Anwender | |
| 1: Anforderungen werden erfüllt | |
| Begründung | Da die Software speziell für Datenmodelltransformationen entworfen wurde, |
| C.1.1 Kompatibilität zu Betriebssystemen | |
| 1: Software ist kompatibel zum Betriebssystem | |
| Begründung | Die Software kann sowohl unter Windows, Linux, als auch einem Mac installiert werden. |
| C.1.2 Spezialisierung der Software | |
| ii: Mehrzweck | |
| Begründung | Die Software ist als abgespeckte Alternative zu FME konzipiert. |
| C.1.3 Mögliche Interaktion mit anderer Software | |
| 0: Interaktionen sind unzureichend möglich | |
| Begründung | Interaktionen sind nur dadurch vorgesehen, dass fremde Formate (z.B. XSLX) verwendet und exportiert werden können. |
| C.1.4 Anwender-Community zum Austausch von Fachkenntnissen | |
| 1: Community ist vorhanden | |

| | |
|---|---|
| Begründung | Es werden offizielle Foren zum Austausch angeboten. Jedoch ist der Austausch in den Foren spärlich. |
| C.2 Entwicklungsstand der Software | |
| C.2.0 Gesamtwertung des Entwicklungsstandes | |
| 0: Entwicklungsstand ist nicht zufriedenstellend | |
| Begründung | Die Software wird möglicherweise derzeit nicht weiterentwickelt. |
| C.2.1 Status der Projektentwicklung | |
| 0: Projekt wird unzureichend weiterentwickelt | |
| Begründung | Der genaue Projektstatus ist unbekannt. |
| C.2.2 Regelmäßigkeit der Softwareaktualisierung | |
| 0: Aktualisierung findet unzureichend statt | |
| Begründung | Die letzte Aktualisierung der Software erfolgte im Jahr 2013. |
| C.3 Kostenstruktur der Software | |
| C.3.0 Gesamtwertung der Kostenstruktur | |
| 1: Es fallen keine Kosten an | |
| Begründung | GeoKettle ist eine Open-Source-Software und somit kostenlos. |
| C.3.1 Anschaffungskosten der Software | |
| 2: Es fallen keine Anschaffungskosten an | |
| Begründung | Die Software kann frei heruntergeladen und installiert werden. |
| C.3.2 Betriebskosten der Software (Lizenzgebühren) | |
| 2: Es fallen keine Lizenzgebühren an | |
| Begründung | Lizenzgebühren beim Betrieb der Software sind nicht zu entrichten. |
| C.3.3 Betriebskosten der Software (Personalstunden) | |
| 0: Personalstunden sind relevant | |
| Begründung | Der Aufwand bei der Gestaltung, Durchführung und Initialisierung von Transformationen ist enorm. |
| C.3.4 Lebenszyklus der Software | |
| 1: Lebenszyklus ist hoch durch fehlende Bindung an Lizenzen | |
| Begründung | Da keine kostenpflichtige Lizenz erforderlich ist, werden Aktualisierungen in die bestehende Edition eingebunden. Wird eine neue Edition entwickelt, kann diese wiederum kostenfrei bezogen werden. |

Tab. 6.8 – GeoKettle: Bewertung der Software nach Bewertungskatalog C: Sonstige Kategorien

6.5.4 Abschließende Bewertung von GeoKettle

GeoKettle ist nur bedingt für Datenmodelltransformationen geeignet. Die Gründe sind den Bewertungskatalogen und Kapitel 5 zu entnehmen. Eindeutig hat GeoKettle Probleme mit den Datenformaten der vorliegenden Anwendungsfälle. Mit der Verwendung des Formats Shapefile könnten jedoch Transformationen mit Einschränkungen gelingen.

6.6 Bewertung von HUMBOLDT Alignment Editor

HALE wird in den nachgestellten Abschnitten nach den entwickelten Katalogen evaluiert und bewertet. Es wird zu jedem identifizierten Kriterium die Wertung angegeben und eine Begründung angeführt, weshalb die Wertung gewählt wurde. Die Bewertung bezieht sich auf alle drei Transformationsfälle zusammen.

6.6.1 Bewertung von HALE nach Bewertungskatalog A

| Katalog A: Kriterien zur Evaluierung der Ergebnisse | |
|--|--|
| A.0 Gesamtwertung der Evaluierung der Ergebnisse | |
| 0: Evaluierung stellt bedingt durch Fehler ungenügende oder mangelhafte Resultate fest | |
| Begründung | Die Ergebnisse sind in den Transformationsfällen I und II valide und wohlgeformt und die Objektorientierung wird in den Zieldaten korrekt hergestellt. Im Transformationsfall III konnte die Geometrie der Gebäudehülle nicht übertragen werden. |
| A.1 Prüfung der Wohlgeformtheit zum Zielmodell | |
| 1: Daten sind wohlgeformt | |
| Begründung | Die Wohlgeformtheit wurde in allen Transformationsfällen festgestellt. |
| A.2 Prüfung auf Validierung zum Zielmodell | |
| 0: Daten sind nicht valide | |
| Begründung | In den Transformationsfällen I und II liefert HALE valide Ergebnisse. Im Transformationsfall III konnte die Geometrie der Gebäudehülle nicht übertragen werden. |
| A.3 Prüfung auf semantische Vollständigkeit | |
| 1: Daten sind semantisch vollständig | |
| Begründung | Es sind alle nicht geometrischen Inhalte vorhanden. Jedoch wurden teilweise neue <code>gml:id</code> erzeugt, anstatt die originären Attributwerte zu übertragen. |
| A.4 Prüfung der geometrischen Transformation | |
| 0: Geometrie ist nicht korrekt | |
| Begründung | In Transformationsfall III fehlt die Geometrie der Gebäudehülle. Hingegen wurde die <code>TerrainIntersection</code> korrekt übertragen. |
| A.5 Prüfung der Koordinatenreferenz-Transformation | |
| 1: Koordinatenreferenz ist korrekt | |
| Begründung | Die Reprojektierung der Koordinatenreferenzsysteme wurde strichprobenartig als korrekt befunden. |
| A.6 Prüfung der Identifikator-Zuweisung | |
| 0: Objektidentifikatoren sind nicht korrekt | |
| Begründung | Die <code>gml:ids</code> werden nach korrekter Kodierung erzeugt, allerdings anstatt die originären Werte zu übernehmen. Eine weitere Erzeugung von Identifikatoren mit Funktionen war nicht notwendig. |

| | |
|--|--|
| A.7 Prüfung der Kodierung | |
| 1: Kodierung ist korrekt | |
| Begründung | Die Kodierung der resultierenden Datensätze ist korrekt. |
| A.8 Prüfung der Objekthierarchien | |
| 1: Objektorientierung ist korrekt | |
| Begründung | HALE nimmt keine Abwandlung der Objektorientierung vor. Diese wird im Quellschema mit eingelesen und durch das Zielschema erzeugt. |

Tab. 6.9 – FME: Bewertung der Software nach Bewertungskatalog A: Evaluierung der Ergebnisse

6.6.2 Bewertung von HALE nach Bewertungskatalog B

| | |
|---|--|
| Katalog B: | Leistungsvermögen der Software |
| B.1 Transformationsansatz der Software | |
| B.1.0 Gesamtwertung des Transformationsansatzes | |
| 1: Transformationsansatz ist ausreichend | |
| Begründung | Nach Betrachtung aller Wertungen kann ein ausreichender Transformationsansatz attestiert werden. |
| B.1.1 Systemabhängigkeit | |
| 1: Systemabhängigkeit ist nicht gegeben | |
| Begründung | HALE ist eine Open-Source-Software und frei verwendbar. |
| B.1.2 Mögliche Abbildungsebenen | |
| 2: Transferformatebene (Abbildung) und Datenebene (Anwendung) | |
| Begründung | Die Transformationsregeln werden zwischen den Transferformaten definiert und anschließend auf eingelesene Datensätze zur Transformation angewandt. Die Transformationsregeln können zudem exportiert werden (z.B CSV). |
| B.1.3 Initialisierbare Richtungen der Transformation | |
| i: undirektional; iii: horizontal | |
| Begründung | Die Transformationen werden fix zwischen Quell- und Ziel-Anwendungsschema der gleichen Ebene definiert. |
| B.1.4 Arten der möglichen Transformationsausführung | |
| i: offline | |
| Begründung | Als Desktop-Anwendung führt HALE die Transformationen offline aus. |
| B.1.5 Skalierbarkeit der Transformation | |
| Begründung | Die Skalierbarkeit ist nur durch Beeinflussung der Größe des Quelldatensatzes möglich. Je größer der Datensatz, desto beanspruchender ist die Transformation. |
| B.1.6 Mögliche Strukturen zur Ablaufsteuerung | |

| | |
|---|---|
| Begründung | Es können nur die nötigsten Einstellungen getroffen werden. Die Software stellt eine GUI zur Verfügung, die optimal auf Datenmodelltransformationen zugeschnitten ist. |
| B.2 Expressivität der verwendeten Abbildungssprache | |
| B.2.0 Gesamtwertung der Expressivität der Abbildungssprache | |
| 1: Abbildungssprache ist geeignet | |
| Begründung | HALE bedient sich der Abbildungssprache EDOAL/gOML. Die Sprache wird auch von Beare et al. (2010, S.22f.) in der engeren Auswahl für Datenmodelltransformationen geführt. |
| B.2.1 Repräsentation der Abbildungssprache | |
| 1: Abbildungssprache weist vorwiegend vorteilhafte Eigenschaften auf | |
| Begründung | Die Abbildungssprache wurde auch speziell für Datenmodelltransformationen entworfen. Als Nachteil wird von Beare et al. (2010, S.22f.) angeführt, dass es sich dabei nicht um einen offiziellen Standard handelt und die Entwicklung noch nicht abgeschlossen ist. Beides trifft auch noch zum Zeitpunkt der Abgabe dieser Arbeit zu. |
| B.2.2 Paradigma der Abbildungssprache | |
| 1: Sprachparadigma der Abbildungssprache wirkt sich kaum bis nicht nachteilig aus | |
| Begründung | Die Abbildungssprache weist ein deklaratives Sprachparadigma auf. Es konnten keine unmittelbar nachweisbaren Nachteile bei der Verwendung von gOML festgestellt werden. |
| B.2.3 Handhabung anderer Sprachparadigmen | |
| 1: Abbildungssprache ist kompatibel | |
| Begründung | Die Abbildungssprache ist in der Lage, die objektorientierten Strukturen der Anwendungsschemata respektive Datensätze fehlerfrei zu handhaben. |
| B.2.4 Versionskompatibilität zu Eingangsdaten und -schemata | |
| 1: Abbildungssprache ist kompatibel | |
| Begründung | Alle Datensätze und Schemata konnten in den vorliegenden Versionen sowohl gelesen als auch geschrieben werden. |
| B.3 Transformationskompetenz der Software | |
| B.3.0 Gesamtwertung der Transformationskompetenz der Software | |
| 0: Transformationskompetenz ist ungenügend | |
| Begründung | HALE lieferte für die Transformationsfälle I und II die besten Ergebnisse, scheiterte jedoch in Transformationsfall III an der Übertragung der Geometrie der Gebäudehüllen. |
| B.3.1 Datenmanagement | |
| 1: Datenmanagement ist für den Nutzer nur bedingt zugänglich | |

| | |
|---|---|
| Begründung | Das Datenmanagement ist schlicht: Es werden die Anwendungsschemata und der zu transformierende Datensatz eingeladen. Die Schemata und Datensätze sind nicht weiter manipulierbar. Das Berichtswesen gibt nur unzureichend Informationen dazu wieder. |
| B.3.2 Aktualität der Funktionalität | |
| 1: Software wird via Updates aktuell gehalten | |
| Begründung | Alle Schemata und Datensätze zu den vorliegenden Anwendungsfällen konnten gelesen und verarbeitet werden. In HALE werden regelmäßig Updates mit neuen Funktionalitäten und Anpassungen eingepflegt. |
| B.3.3 Erweiterbarkeit der Abbildungssprache | |
| 0: Abbildungssprache ist nicht erweiterbar | |
| Begründung | Die Abbildungssprache ist in HALE unzugänglich hinterlegt. Anpassungen können nur durch den Anbieter vorgenommen werden. |
| B.3.4 Berichtswesen | |
| 0: Dokumentationsvorgänge sind unzureichend vorhanden | |
| Begründung | Es wird ein Report zum Transformationsvorgang erzeugt, der jedoch nur bedingt Erkenntnisse liefert. Fehlermeldungen sind nicht ausreichend spezifiziert, um als Anwender die Fehlerursache begründen zu können. |
| B.3.5 Qualitätsmanagement | |
| 0: Mechanismen sind unzureichend vorhanden | |
| Begründung | Das Resultat einer Transformation kann während des Vorgangs (und nur währenddessen) validiert werden. Das Ergebnis der Validierung war in allen Fällen negativ (ohne Angabe von Gründen), weshalb der Validierungsmechanismus in Frage gestellt werden muss. |
| B.3.6 Vorhandene Spezialisierungen für Fachanwendungen | |
| 1: Spezialisierungen für Anwendungsfälle sind vorhanden | |
| Begründung | HALE bietet u.a. Codelists an, die jedoch ausschließlich auf INSPIRE abzielen. |
| B.4 Funktionsbibliothek der Software | |
| B.4.0 Gesamtwertung der Funktionsbibliothek | |
| 1: Funktionsbibliothek ist ausreichend | |
| Begründung | Es konnten prinzipiell alle Transformationen durchgeführt werden, jedoch sind die Funktionen zur Übertragung der Geometrie zu prüfen. |
| B.4.1 Umfang der Funktionsbibliothek | |
| 1: Funktionsbibliothek beinhaltet die benötigten Konstrukte | |
| Begründung | Die Funktionsbibliothek ist schlicht gehalten, genügte jedoch den Anforderungen aus den Anwendungsfällen. Der Fehler in der Übertragung der Geometrie der Gebäudehüllen aus CityGML 1.0 ist vermutlich nicht durch das Fehlen einer entsprechenden Funktion begründet, sondern in deren Arbeitsweise. |
| B.4.2 Funktionen zur Bearbeitung der Geometrie | |

| | |
|---|---|
| 0: Geometrie kann nicht verarbeitet werden | |
| Begründung | Die Geometrien aus den Transformationsfällen I und II sowie die <i>TerrainIntersection</i> wurde erfolgreich in den Zieldatensätzen erzeugt. Die Gebäudehüllen aus CityGML 1.0 fehlen hingegen. |
| B.4.3 Funktionen zur Bearbeitung von Inhalten | |
| 1: Inhalte können verarbeitet werden | |
| Begründung | Es konnten alle vergebenen Inhalte generiert werden (sogar aus «CI_Citation»). Eine Erweiterung von Inhalten über die Elemente des Zielschemas hinaus ist nicht möglich. |
| B.4.4 Initialisierung von Bedingungen und Schleifen | |
| 1: Bedingungen und Schleifen sind unzureichend vorhanden | |
| Begründung | Es kann nur eine bedingte Wertbelegung auf Basis eines ausgewählten Quell-FeatureTypes erfolgen. Weitere Bedingungen sind in Groovy Script umzusetzen, wurden im Zuge dieser Arbeit aber nicht geprüft. Der Anwender muss über die entsprechende Programmierkenntnis verfügen. |
| B.4.5 Qualität der Funktionsbibliothek | |
| 1: Qualität ist ausreichend | |
| Begründung | Die Funktionen funktionieren gut, sind ausreichend dokumentiert und ohne Aufwand oftmals intuitiv bedien- und arrangierbar. Für komplexere Funktionen, wie z.B. <i>XMLQuery</i> , ist jedoch Fachkenntnis nötig. |
| B.4.6 Erweiterbarkeit mit eigenen Funktionen | |
| 0: Funktionsbibliothek ist nicht oder unzureichend erweiterbar | |
| Begründung | Es können keine eigenen neuen Funktionen in die Bibliothek integriert werden. |
| B.5 Benutzerfreundlichkeit der Software | |
| B.5.0 Gesamtwertung der Benutzerfreundlichkeit | |
| 1: Benutzerfreundlichkeit ist ausreichend | |
| Begründung | Durch die Spezialisierung auf Datenmodelltransformationen ist die Benutzerfreundlichkeit in besonderem Maß gegeben. |
| B.5.1 Form der Abbildungsfestlegung und Durchführung | |
| 1: Abbildung wird durch grafische Benutzerschnittstelle unterstützt | |
| Begründung | HALE stellt eine schlichte GUI mit diversen Ansichtsfenstern zur Verfügung. Die Funktionen werden nach Markierung der beteiligten Attribute aus einem Menü gewählt, wobei von HALE schon eine Vorauswahl an Funktionen vorgegeben wird. Nach Erzeugung einer Abbildung kann diese nachträglich editiert werden. Der Anwender wird von HALE mittels Steuerungsdialogen durch den Vorgang Transformation geführt. |
| B.5.2 Mögliche Anwenderführung | |
| 1: Anwender wird durch Dialogsteuerung geführt | |

| | |
|--|---|
| Begründung | Bei der Ausführung der Transformation spult HALE eine Abfolge von Steuerungsdialogen ab, in denen die notwendigen Einstellungen dazu getätigt werden können. |
| B.5.3 Grad des benötigten Know-hows zur Anwendung | |
| 1: Fachkenntnisse sind in Fachanwendung erforderlich | |
| Begründung | Fachkenntnisse im Umgang mit Transferformaten, Datenformaten etc. sollten grundlegend vorhanden sein. Programmierkenntnisse sind für komplexere Vorgänge mit Groovy Script erforderlich. Die reine Bedienung der Software ist auch für geübtes Personal ohne Fachkenntnisse in der Geoinformatik nach Einarbeitung möglich. |
| B.5.4 Ausmaß der möglichen Automatisierung | |
| 0: Automatisierung ist nicht möglich | |
| Begründung | Jeder Transformationsvorgang muss in HALE separat ausgeführt werden. Dazu müssen die Quellen zu den Schemata und Datensätzen jeweils manuell angepasst werden. Die Einstellungen in den Steuerungsdialogen kann man nicht speichern, um den Vorgang zu beschleunigen. Es konnte auch keine Kenntnis darüber gewonnen werden, dass HALE über externe Skripte ansteuerbar wäre. |
| B.5.5 Dokumentation | |
| 1: Dokumentation ist unzureichend vorhanden | |
| Begründung | Es wird ein Report über den Transformationsvorgang erstellt. Dieser liefert allerdings nur unzureichende Informationen und bietet keine ausreichende Hilfestellung zur Identifikation von Fehlerquellen an. |
| B.5.6 Unterstützung durch Support | |
| 1: Support wird angeboten | |
| Begründung | Der Support kann per E-Mail kontaktiert werden. Über eine Recherche kann auch Fachpersonal identifiziert werden. |
| B.5.7 Sonstige zusätzliche Funktionen | |
| 1: Sonstige Funktionen sind gegeben | |
| Begründung | HALE bietet einen Viewer, in dem Datensätze in einer georeferenzierten Kartenansicht betrachtet werden können. Das Koordinatenreferenzsystem ist jedoch vorgegeben, da die Kartengrundlage aus Open Street Map herangezogen wird. |
| B.6 Schnittstellen der Software | |
| B.6.0 Gesamtwertung zu den Schnittstellen | |
| 1: Schnittstellen sind ausreichend | |
| Begründung | Es sind Schnittstellen eingebunden, durch die eine Kommunikation mit Web Services möglich ist. Es können auch Dienste wie z.B. WFS aufgerufen und Daten aus externen Quellen herangezogen werden. |
| B.6.1 Enthaltene Schnittstellen | |
| 1: Schnittstellen zum Import/Export sind vorhanden | |

| | |
|---|---|
| Begründung | Die Anwendungsschemata und Datensätze können aus diversen Quellen eingeladen werden (u.a. WFS, URL, Datenbank). Außerdem können getätigte Abbildungen exportiert werden (z.B. im Format CSV oder XSLX). |
| B.6.2 Kompatibilität mit Web Services | |
| 1: Schnittstellen sind kompatibel zu aktuellen Standards für Web Services | |
| Begründung | Die Interaktion mit Web Services (z.B. WFS) ist nach aktuellen Standards möglich. |
| B.6.3 Routinen zum Einlesen von Datenformaten | |
| 1: Routinen sind vorhanden | |
| Begründung | Eine Routine steht zur Verfügung. Es können Netzdienste (z.B. WFS) und externe Quellen eingebunden werden. Die unterstützten Formate sind hingegen begrenzt. |
| B.6.4 Routinen zum Export von Datenformaten | |
| 1: Routinen sind vorhanden | |
| Begründung | Eine Routine ist vorhanden. Es kann auch in einen WFS-T exportiert werden. |

Tab. 6.10 – HALE: Bewertung der Software nach Bewertungskatalog B: Leistungsvermögen der Software

6.6.3 Bewertung von HALE nach Bewertungskatalog C

| | |
|--|---|
| Katalog C: | Sonstige Kategorien |
| C.1 Anforderungen durch den Anwender | |
| C.1.0 Gesamtwertung der spezifischen Anforderungen durch den Anwender | |
| 1: Anforderungen werden erfüllt | |
| Begründung | Da die Software speziell für Datenmodelltransformationen entworfen wurde, wird sie auch weitestgehend die Anforderungen durch den Anwender erfüllen. |
| C.1.1 Kompatibilität zu Betriebssystemen | |
| 1: Software ist kompatibel zum Betriebssystem | |
| Begründung | Die Software kann sowohl unter Windows, Linux, als auch einem Mac installiert werden. |
| C.1.2 Spezialisierung der Software | |
| i: spezialisiert | |
| Begründung | Die Software ist speziell für Datenmodelltransformationen konzipiert worden. |
| C.1.3 Mögliche Interaktion mit anderer Software | |
| 0: Interaktionen sind unzureichend möglich | |
| Begründung | Interaktionen sind nur dadurch vorgesehen, dass fremde Formate (z.B. XSLX) verwendet und exportiert werden können. Für FME wurde ein HALE Plug-in entwickelt, dass jedoch in dieser Arbeit nicht geprüft wurde. |
| C.1.4 Anwender-Community zum Austausch von Fachkenntnissen | |
| 2: Community ist vorhanden und aktiv | |

| | |
|---|---|
| Begründung | Es werden offizielle Foren zum Austausch angeboten. Entwickler von FME sind in den Foren ebenfalls aktiv. |
| C.2 Entwicklungsstand der Software | |
| C.2.0 Gesamtwertung des Entwicklungsstandes | |
| 1: Entwicklungsstand ist zufriedenstellend | |
| Begründung | Die Software wird fortlaufend weiterentwickelt und durch Updates aktuell gehalten. |
| C.2.1 Status der Projektentwicklung | |
| 2: Projekt wird fortlaufend weiterentwickelt | |
| Begründung | Es erscheinen regelmäßig Updates zur Software, mit denen auch neue Funktionalitäten eingebunden werden. |
| C.2.2 Regelmäßigkeit der Softwareaktualisierung | |
| 2: Aktualisierung findet regelmäßig statt | |
| Begründung | Die Aktualisierung der Software folgt keiner bestimmten Regelmäßigkeit. |
| C.3 Kostenstruktur der Software | |
| C.3.0 Gesamtwertung der Kostenstruktur | |
| 1: Es fallen keine Kosten an | |
| Begründung | HALE ist eine Open-Source-Software und somit kostenlos. |
| C.3.1 Anschaffungskosten der Software | |
| 2: Es fallen keine Anschaffungskosten an (freie Software) | |
| Begründung | Die Software kann frei heruntergeladen und installiert werden. |
| C.3.2 Betriebskosten der Software (Lizenzgebühren) | |
| 2: Es fallen Lizenzgebühren an | |
| Begründung | Lizenzgebühren beim Betrieb der Software sind nicht zu entrichten. |
| C.3.3 Betriebskosten der Software (Personalstunden) | |
| 1: Personalstunden sind vernachlässigbar gering | |
| Begründung | Der Aufwand bei der Durchführung hält sich in Grenzen und kann durch Fachpersonal ohne großen Aufwand erfolgen. Da die Transformationsregeln wiederverwendet werden können sind diese nur nach Änderungen in den Anwendungsschemata anzupassen. |
| C.3.4 Lebenszyklus der Software | |
| 1: Lebenszyklus ist hoch durch fehlende Bindung an Lizenzen | |
| Begründung | Da keine kostenpflichtige Lizenz erforderlich ist, werden Aktualisierungen in die bestehende Edition eingebunden. Wird eine neue Edition entwickelt, kann diese wiederum kostenfrei bezogen werden. |

Tab. 6.11 – HALE: Bewertung der Software nach Bewertungskatalog C: Sonstige Kategorien

6.6.4 Abschließende Bewertung von HALE

HALE ist als Software für Datenmodelltransformationen sehr gut geeignet. Jedoch sind möglicherweise noch Fehler enthalten, wodurch nicht alle in Betracht kommenden Geometrien (Gebäudehüllen: *boundarySurfaces* verarbeitet werden können. Bislang werden auch alle Elemente und Informationen aus den originären Daten unterhalb des Levels des Geometrieobjekts (z.B. *gml:ids* der *Sub-Surfaces*) bei der Transformation nicht berücksichtigt [ESDI Community 2015]. Speziell für die Anwendung von CityGML besteht demnach noch Bedarf an Nachbesserung, so dass auch im Profil Core3D die Gebäudehüllen übertragen werden können. Die Einarbeitung in HALE ist zügig möglich. Die Transformationsvorgänge werden streng geführt und erleichtern dadurch die Handhabung der Software. Allerdings hat dies auch Einschränkungen in der Reduzierung des Arbeitsaufwands zur Folge, da die Schritte und Angaben bei jeder Ausführung einer Transformation stets zu wiederholen sind. Zudem werden Automatisierungen dadurch erschwert. Ein besonderer Vorteil ist, dass das ETL-Tool fortlaufend weiterentwickelt wird, es frei erhältlich ist und der Anwender keine finanziellen Mittel dafür aufbringen muss.

6.7 Vergleichende Wertung

Zum Vergleich der Tools werden die Bewertungsschlüssel aus den Wertungen herangezogen. Dazu werden die vorangestellten Angaben in Form von Zahlen aus den Bewertungsschlüsseln in den Tabellen innerhalb jeder Kategorie aufaddiert. Es konnten je nach Wertung eines Kriteriums 0 bis 2 Punkte erzielt werden (die möglichen Punktwertungen sind in den Katalogen der Abschnitte 6.2.2 und 6.2.3 aufgelistet). In Tabelle 6.12 sind die Wertungen nach den Bewertungskategorien und insgesamt eingefügt. In Klammern hinter der erzielten Punktwertung ist die mögliche Gesamtpunktzahl zu jeder Kategorie und insgesamt angegeben.

| Kategorie | FME | GeoKettle | HALE |
|-----------|---------|-----------|---------|
| A | 9 (9) | 2 (9) | 5 (9) |
| B.1 | 3 (4) | 1 (4) | 4 (4) |
| B.2 | 5 (6) | 0 (6) | 5 (6) |
| B.3 | 8 (8) | 3 (8) | 3 (8) |
| B.4 | 12 (12) | 2 (12) | 5 (12) |
| B.5 | 9 (11) | 4 (11) | 7 (11) |
| B.6 | 5 (5) | 3 (5) | 5 (5) |
| C.1 | 5 (6) | 7 (6) | 4 (6) |
| C.2 | 5 (5) | 0 (5) | 5 (5) |
| C.3 | 4 (8) | 6 (8) | 7 (8) |
| insgesamt | 65 (74) | 28 (74) | 50 (74) |

Tab. 6.12 – Punktwertung der ETL-Tools im Vergleich

Die beste Wertung erzielte demnach FME, während GeoKettle eine schlechte Wertung zugewiesen bekommt. Es wird auch deutlich, in welchen Rubriken die Tools ihre Stärken und Schwächen haben. Eine offensichtliche Stärke der Tools GeoKettle und HALE liegt in der Kostenstruktur, da die Tools frei zu beziehen sind. GeoKettle zeigt vor allem Schwächen in den Kategorien, in denen die Transformationsansätze und -möglichkeiten bewertet werden. Obwohl HALE in der Wertung hinter FME abfällt, ist die Software für manche Anwendungsfälle dennoch als mindestens ebenbürtig anzusehen. Die Unterschiede zwischen beiden Tools ergeben sich vornehmlich durch die umfangreicheren Möglichkeiten von FME, beispielsweise bedingt durch die deutlich umfassendere Funktionsbibliothek, deren Möglichkeiten in der vorliegenden Arbeit allerdings nicht ausgeschöpft werden mussten. Würde man diese Vorzüge außer Acht lassen, wären beide Tools letztlich in der Wertung gleich auf und als gleichermaßen gut einzustufen.

Kapitel 7

Schlussfolgerung und Ausblick

7.1 Zusammenfassung der Ergebnisse und Schluss

Diese Arbeit liefert einen Beitrag zur Realisierung von INSPIRE in der GDI-DE. An einigen Stellen wurde aber auch darüber hinaus gedacht und die Machbarkeit von Datenmodelltransformationen im weiteren Kontext und die Grenzen der Darstellung von Abbildungsregeln in Tabellen erörtert.

Es wurden mit ausgewählten Anwendungsfällen und drei in Frage kommenden ETL-Tools Datenmodelltransformationen durchgeführt, wie sie zukünftig auch von den Datenhaltern der Geodateninfrastrukturen durchzuführen sind. Dazu wurden durch das LDBV und die AdV Abbildungstabellen bereitgestellt. Aus den Abbildungstabellen konnten auch Anforderungen durch die Anwendungsfälle an die ETL-Tools identifiziert werden.

Durch die Transformationen wurden sowohl valide und gebrauchsfähige Datensätze hergestellt, als auch einige Fehlerquellen detektiert. Die aufgedeckten Fehler sind letztlich vielfältig und liegen oftmals im Detail. Besonders hervorzuheben ist, dass es HALE nicht möglich war, die Geometrie aus CityGML 1.0 im Zielschema BuildingsCore3DLoD2 zu verarbeiten, so dass diese trotz weitestgehender Versuche nicht in die Zieldaten geschrieben werden konnte. Das Tool GeoKettle zeigte sich für die Anwendung aller ausgewählten Transformationsfälle als ungeeignet. Zudem konnten auch keine validen Datensätze damit generiert werden.

Es wurde auch der interne Arbeitsablauf von FME eingehender untersucht. Dabei wurden die relationalen Strukturen identifiziert, derer sich FME bedient, und das FME-eigene Datenmodell umschrieben. Als Erkenntnis konnte daraus abgeleitet werden, wie die Paradigmenwechsel von objekt-orientiertem Paradigma der Daten und relationalem Paradigma von FME vonstatten gehen.

Anschließend wurde ein umfangreiches Konzept zur Bewertung der Datenmodelltransformationswerkzeuge auf Basis von 51 Bewertungskriterien entwickelt und beschrieben. Die Bewertungsgrundlage speiste sich vor allem aus den Ergebnissen der durchgeführten Datenmodelltransformationen. Zuzüglich wurden aber auch Erkenntnisse aus anderen Arbeiten berücksichtigt.

Die Bewertung erfolgte in drei Bewertungskatalogen für jedes der drei ETL-Tools. Abschließend wur-

den die Wertungen verglichen und unter Hinzuziehung der Erkenntnisse und Ergebnisse aus Kapitel 5 die Eignung oder Nicht-Eignung der Tools festgestellt.

7.2 Fazit und Empfehlungen

FME und HALE können beide zur Anwendung für Datenmodelltransformationen empfohlen werden, wenngleich noch Fehler in den Tools zu beheben sind.

In den Transformationsfällen I & II konnten für FME keine Fehler festgestellt werden. Dennoch ging aus Popp (2015) hervor, dass die Abbildung von Geometrien unter Umständen nicht in allen Fällen reibungslos respektive korrekt stattfindet.

GeoKettle konnte sich nicht für die Anwendung im Kontext von INSPIRE empfehlen, insbesondere auch dann, wenn mit Transferformaten gearbeitet wird. Für Anwendungsfälle, die keine komplexen Strukturen beinhalten, könnten mit Verwendung von Shapefiles als In- und Output-Format valide Ergebnisse erzielt werden.

Für die Transformation zu INSPIRE scheint HALE gut geeignet, da die Implementierung im besonderen Maß auf die Initiative abzielt. Treten jedoch in Anwendungsfällen, insbesondere auch außerhalb von INSPIRE, komplexere Konstrukte und Geometrien in Erscheinung, ist HALE, bedingt durch seine limitierte Funktionsbibliothek, nicht mehr ausreichend gut gewappnet.

FME bietet deutlich mehr Möglichkeiten als HALE, ist jedoch eine kostenpflichtige Software. Da der Support gut zu erreichen ist und auf die Anregungen der Anwender eingeht, können aufgedeckte Fehler gegebenenfalls sehr schnell behoben werden. Auch im Zuge dieser Arbeit wurde der Support kontaktiert und es wurde sofort klar gestellt, dass mögliche Fehler umgehend untersucht werden. Für die Anwendung mit Transformationsfällen aus INSPIRE ist FME gleichermaßen gut geeignet wie HALE. Für die Transformation von CityGML scheint FME nach derzeitigem technischen Stand besser geeignet zu sein, als HALE.

7.3 Ausblick auf weitere Fragestellungen

Die vorliegende Arbeit behandelt vornehmlich Anwendungsfälle mit einem Zielschema aus INSPIRE. Da aber die Datenmodelltransformation zu INSPIRE gegenwärtig und auch zukünftig eine wichtige Rolle in den nationalen Geodateninfrastrukturen einnimmt, können die Datenanbieter wichtige Erkenntnisse aus den Ergebnissen dieser Thesis erlangen. Es ist auch davon auszugehen, dass manche der untersuchten Datenmodelltransformationswerkzeuge im Hinblick auf INSPIRE und die zunehmende Digitalisierung im Allgemeinen weiterentwickelt werden. Möglicherweise kann diese Arbeit auch dazu beitragen.

Das Spektrum von INSPIRE ist weitaus umfassender, als es die ausgewählten Anwendungsfälle

abdecken können. Entsprechend ist auch davon auszugehen, dass noch viele weitere Frage- und Problemstellungen geklärt werden müssen, bevor INSPIRE zu einem obligatorischen Bestandteil der Geodateninfrastrukturen werden kann. Sowohl die gewählten Anwendungsfälle, als auch die verwendeten Datenmodelltransformationswerkzeuge, stellen nur eine Auswahl aus zahlreichen weiteren Möglichkeiten dar. Andere Tools bieten möglicherweise eine praktikablere Alternative, was jedoch weiterhin zu untersuchen wäre.

Die Gründe, welche dazu führten, dass die hier beschriebenen Tools auch Fehler verursachten, konnten nicht hinreichend aufgedeckt werden. Insbesondere die Verarbeitung von Geometrien in HALE sollte näher untersucht werden.

Aus den hier vorliegenden Anwendungsfällen konnten nur unzureichend Erkenntnisse darüber gewonnen werden, wie die ETL-Tools bei der Anwendung mit Anwendungsfällen außerhalb von INSPIRE abschneiden. Um eine umfassendere Bewertung der Tools hinsichtlich ihrer Eignung für semantische Datenmodelltransformationen im Allgemeinen vornehmen zu können, wären weitere Untersuchungen mit zusätzlichen Anwendungsfällen aus der Geoinformatik nötig.

Da FME auch extern durch Python angesteuert werden kann, könnten in weiteren Forschungsarbeiten Ansätze entwickelt werden, Datenmodelltransformationen in FME vollständig zu automatisieren. Möglicherweise könnte auch HALE dahingehend weiterentwickelt werden.

Literaturverzeichnis

- [AdV CityGML 2014] Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (AdV): AdV-CityGML-Profile für 3D-Gebäudemodelle der AdV, GeoInfoDok 6.0, 08.12.2014. URL <http://www.adv-online.de/icc/extdeu/nav/a99/binarywriterservlet?imgUid=01e607e7-a797-2a41-b71d-b35472e13d63uBasVariant=11111111-1111-1111-1111-111111111111> - abgerufen am 06. Mai 2015
- [BayGDIG 2008] Bayerisches Geodateninfrastrukturgesetz. Bayerische Staatsregierung, 2008. URL <http://www.gesetze-bayern.de/jportal/portal/page/bsbayprod.psml?showdoccase=1doc.id=jlr-GDIGBYrahmendoc.part=X> - abgerufen am 19. April 2015.
- [BLfD 2015] Bayerisches Landesamt für Denkmalpflege: Homepage. URL <http://www.blfd.bayern.de/> - abgerufen am 19. Juni 2015.
- [CityGML 2015] CityGML: Homepage für CityGML. URL <http://www.citygml.org/> - abgerufen am 21. August 2015.
- [CityGML Version 1.0] Open Geospatial Consortium (Hrsg.): CityGML UML diagrams - CityGML Encoding Standard Version 1.0.0, OGC Doc. No. 08-007r1. Open Geospatial Consortium, 2008. URL http://www.citygml.org/fileadmin/citygml/docs/CityGML_1_0_0_UML_diagrams.pdf - abgerufen am 06. Mai 2015.
- [CityGML Version 2.0] Open Geospatial Consortium (Hrsg.): OGC City Geography Markup Language Encoding Standard Version 2.0.0, OGC Doc. No. 12-019. Open Geospatial Consortium, 2013. URL <http://www.opengeospatial.org/docs/is> - abgerufen am 25. September 2015.
- [DIN 2005] DIN Deutsches Institut für Normung e.V. (Hrsg.): DIN EN ISO 19107: Geoinformation - Raumbezugsschema (ISO 19107:2003). Englische Fassung EN ISO 19107:2005. Berlin: Beuth, 2005.
- [DIN 2007] DIN Deutsches Institut für Normung e.V. (Hrsg.): DIN EN Iso 19136: Geoinformation - Geography Markup Language (GML) (ISO 19136:2007). Englische Fassung EN ISO 19136:2009. Berlin: Beuth, 2009.
- [DIN 2015] DIN Deutsches Institut für Normung e.V. (Hrsg.): DIN EN ISO 19101: Geoinformation - Referenzmodell - Teil 1: Grundsätze (ISO 19101-1:2014). Englische Fassung EN ISO 19101-1:2014. Berlin: Beuth, 2015.

- [Donaubauer 2004] Donaubauer, Andreas: Interoperable Nutzung verteilter Geodatenbanken mittels standardisierter Geo Web Services. München: Technische Universität München, Dissertation, 2004.
- [Duden 2015] Duden online. 2015. URL <http://www.duden.de/> - abgerufen am 19. Juni 2015.
- [ESDI Community 2015] ESDI Community Forum für HALE. URL <http://www.esdi-community.eu/boards/3/topics/325> - abgerufen am 20. September 2015.
- [ESDIN 2015] European Spatial Data Infrastructure with a best practice Network - project community: Homepage. URL <http://www.esdi-community.eu/projects/hale> - abgerufen am 29. Mai 2015.
- [EU Charta 2010] Europäische Union: Konsolidierte Verträge - Charta der Grundrechte der Europäischen Union. Luxemburg: Amt für Veröffentlichungen der Europäischen Union, 2010. URL http://europa.eu/pol/pdf/consolidated-treaties_de.pdf - abgerufen am 16. Juni 2015.
- [Europäisches Parlament und Rat 2007] Europäisches Parlament und Rat der Europäischen Union: Richtlinie 2007/2/EG des Europäischen Parlaments und Rates vom 17. März 2007 zur Schaffung einer Geodateninfrastruktur in der Europäischen Gemeinschaft (INSPIRE). In: Amtsblatt der Europäischen Union 50 (2007), Nr. 108, 25.04.2007. URL <http://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32007L0002from=DE> - abgerufen am 04. März 2015
- [EU-Vertrag 2007] Europäische Union: Vertrag von Lissabon zur Änderung des Vertrags über die Europäische Union und des Vertrags zur Gründung der Europäischen Gemeinschaft. In: Amtsblatt 2007/C 306/01 der Europäischen Union vom 17. Dezember 2007. URL <http://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=OJ:C:2007:306:FULLfrom=DE> - abgerufen am 16. Juni 2015.
- [Fichtinger et al. 2010] Fichtinger, Astrid; Rix, Joachim; Schäffler, Ulrich; Michi, Ines; Gonen, Moses; Reitz, Thorsten: Data Harmonisation Put into Practice by the HUMBOLDT Project. In: *International Journal of Spatial Data Infrastructures* (Article under Review). URL <http://ijmdir.jrc.ec.europa.eu/index.php/ijmdir/article/viewFile/191/248> - abgerufen am 06. April 2015.
- [Fichtinger 2011] Fichtinger, Astrid: Semantische Transformation im Kontext von INSPIRE - dargestellt am Beispiel der grenzüberschreitenden Bodenseeregion. München: Technische Universität München, Dissertation, 2011.
- [FME Workbench Help 2015] Safe Software Inc.: FME Workbench Online Hilfe. URL http://docs.safe.com/fme/2015.0/html/FME_Desktop_Documentation/FME_Desktop_Help.htm?shid=wb_index - abgerufen am 14. Juli 2015.
- [GDI-BY WFS 2013] Geodateninfrastruktur Bayern (GDI-BY): Einrichten von Web Feature Services - Leitfaden. 2013. URL http://www.gdi.bayern.de/file/pdf/922/Leitfaden_WebFeatureServices_1_0.pdf - abgerufen am 23. Juli 2015.

- [GDI-BY 2015] Geodateninfrastruktur Bayern (GDI-BY): Homepage: Was ist eine Geodateninfrastruktur. http://www.gdi.bayern.de/was_ist_gdi.html - abgerufen am 23. Juli 2015.
- [GeoZG 2009] Gesetz über den Zugang zu digitalen Geodaten (Geodatenzugangsgesetz - GeoZG). Bundesministerium der Justiz und für Verbraucherschutz, 2009. URL <http://www.gesetze-im-internet.de/geozg/> - abgerufen am 19. April 2015.
- [Gedrange und Neubert 2008] Gedrange, Claudia; Neubert, Marco: Grenzüberschreitende Homogenisierung von Geobasisdaten. In: Strobl, Josef; Blaschke, Thomas; Griesebner, Gerald (Hrsg.): *Angewandte Geoinformatik* 2008, Beiträge zum 20. AGIT-Symposium Salzburg. Heidelberg: Wichmann, 2008, S. 828 - 837.
- [GeoKettle Dokumentation 2015] Spatialytics GeoKettle Documentation Center: Online Dokumentation von Spatialytics ETL (inkl. GeoKettle). URL http://docs.spatialytics.com/doku.php?id=en:spatialytics_etl - abgerufen am 16. Juni 2015.
- [Geoportal GDI-DE 2015a] Geodateninfrastruktur Deutschland (GDI-DE): Homepage. Informationen zur Direktive INSPIRE. <http://www.geoportal.de/DE/GDI-DE/INSPIRE/Direktive/direktive.html%3bjsessionid=AA0589D6ECBAAF30CB9957D21A3ED6BE?lang=de> - abgerufen am 23. Juli 2015.
- [Geoportal GDI-DE 2015b] Geodateninfrastruktur Deutschland (GDI-DE): Homepage. Informationen zur Durchführungsbestimmung zu den Metadaten. <http://www.geoportal.de/DE/GDI-DE/INSPIRE/Direktive/Metadata/metadata.html?lang=de> - abgerufen am 23. Juli 2015.
- [Geoportal GDI-DE 2015c] Geodateninfrastruktur Deutschland (GDI-DE): Homepage. Informationen zur Durchführungsbestimmung zu den Netzdiensten. <http://www.geoportal.de/DE/GDI-DE/INSPIRE/Direktive/Network-Service/network-service.html?lang=de> - abgerufen am 23. Juli 2015.
- [Geoportal GDI-DE 2015d] Geodateninfrastruktur Deutschland (GDI-DE): Homepage. Informationen zur Durchführungsbestimmung zum Monitoring und Berichtswesen. <http://www.geoportal.de/DE/GDI-DE/INSPIRE/Direktive/Monitoring-und-Reporting/monitoring-und-reporting.html?lang=de> - abgerufen am 23. Juli 2015.
- [Geoportal GDI-DE 2015e] Geodateninfrastruktur Deutschland (GDI-DE): Homepage. Informationen zur Gemeinsamen Nutzung von Daten. <http://www.geoportal.de/DE/GDI-DE/INSPIRE/Direktive/Datasharing/datasharing.html?lang=de> - abgerufen am 23. Juli 2015.
- [Geoportal GDI-DE 2015f] Geodateninfrastruktur Deutschland (GDI-DE): Homepage. Informationen zur Interoperabilität von Geodatensätzen und -diensten. <http://www.geoportal.de/DE/GDI-DE/INSPIRE/Direktive/Data-Specs/data-specs.html?lang=de> - abgerufen am 23. Juli 2015.
- [HALE Forum 2015] HUMBOLDT Alignment Editor official Forum. URL <http://community.esdi-humboldt.eu/boards/3/topics/431> - abgerufen am 16. September 2015.

- [Horák et al. 2011] Horák, Jiří; Juřiková, Lucie; Umlauf, Miroslav; Ježek, Jan; Hanzlová, Markéta; Zlatanová, Sisi: HUMBOLDT Alignment Editor. In: Proceedings Symposium GIS, Ostrava, Tschechische Republik, 23. - 26. Januar 2011. Technische Universität Ostrava.
- [HUMBOLDT project 2010] HUMBOLDT Consortium: HUMBOLDT project Flyer. 2010.
URL http://www.esdi-humboldt.eu/files/1131-humboldt_brochure_2010-projo-001-final.pdf
- abgerufen am 29. Mai 2015.
- [INSPIRE ATS 2008] INSPIRE Drafting Team Data Specifications: INSPIRE D2.3 Definition of Annex Themes and Scope. Drafting Team Data Specifications, Version 3.0, 2008. URL http://inspire.ec.europa.eu/reports/ImplementingRules/DataSpecifications/D2.3_Definition_of_Annex_Themes_and_scope_v3.0.pdf - abgerufen am 06. April 2015.
- [INSPIRE DS PS 2014] INSPIRE Drafting Team Data Specifications: INSPIRE D2.8.I.9 Data Specification on Protected Sites - Technical Guidelines. INSPIRE Thematic Working Group Protected Sites, Version 3.2, 2014. URL http://inspire.ec.europa.eu/documents/Data_Specifications/INSPIRE_DataSpecification_PS_v3.2.pdf - abgerufen am 06. April 2015.
- [INSPIRE DS BU 2013] INSPIRE Drafting Team Data Specifications: INSPIRE D2.8.III.2 Data Specification on Buildings - Technical Guidelines. INSPIRE Thematic Working Group Buildings, Version 3.0, 2013. URL http://inspire.ec.europa.eu/documents/Data_Specifications/INSPIRE_DataSpecification_BU_v3.0.pdf - abgerufen am 06. April 2015.
- [INSPIRE GE 2013] INSPIRE Drafting Team Data Specifications: INSPIRE D2.7 Guidelines for the encoding of spatial data. Drafting Team Data Specifications, Version 3.3, 2013. URL http://inspire.ec.europa.eu/documents/Data_Specifications/D2.7_v3.3rc3.pdf - abgerufen am 06. April 2015.
- [INSPIRE JRC 2014] INSPIRE Drafting Team Data Specifications: INSPIRE D2.5 Generic Conceptual Model. Drafting Team Data Specifications, Version 3.4, 2014. URL http://inspire.ec.europa.eu/documents/Data_Specifications/D2.5_v3.4.pdf - abgerufen am 06. April 2015.
- [INSPIRE MfD 2007] INSPIRE Drafting Team Data Specifications: INSPIRE D2.6 Methodology for the development of data specifications. Drafting Team Data Specifications, Version 2.0 Final, 2007. URL http://inspire.ec.europa.eu/reports/ImplementingRules/inspireDataspD2_6v2.0.pdf - abgerufen am 31. Juli 2015.
- [Kutzner und Eisenhut 2010] Kutzner, Tatjana; Eisenhut, Claude: Vergleichende Untersuchungen zur Modellierung und Modelltransformation in der Region Bodensee im Kontext von INSPIRE. München: Technische Universität München, 2010.

- [Kolbe 2014] Kolbe, Thomas H.: 3D-Geobasisinformationen - Umsetzung und Potenziale. In: AED-SICAD (Hrsg.): Kundentag Bonn 04. Juni 2014, Fachvortrag. URL http://www.aed-sicad.de/as_files/events/2014/Kundentag/03_3D-Geobasisinformationen-Prof.%20Kolbe.pdf - abgerufen am 22. April 2015.
- [Legler und Naumann 2007] Legler, Frank; Naumann, Felix: A Classification of Schema Mappings and Analysis of Mapping Tools. In: Gesellschaft für Informatik. Rheinisch-Westfälische Technische Hochschule Aachen (Hrsg.): 12. GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web. Aachen, 2007. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.231.2990rep=rep1type=pdf> - abgerufen am 19. April 2015.
- [LfU 2015] Bayerisches Landesamt für Umwelt: Homepage. URL <http://www.lfu.bayern.de/natur/schutzgebiete/naturschutzgebiete/index.htm> - abgerufen am 19. Juni 2015.
- [Löwner et al 2014] Löwner, Marc-Oliver; Benner, Joachim; Gröger, Gerhard: Aktuelle Trends in der Entwicklung von CityGML3.0. In: Deutsche Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformatik (Hrsg.): 12. Gemeinsame Jahrestagung der DGfK, der DGPF, der GfGI und des GiN. DGPF Tagungsband 23/2014. URL http://www.researchgate.net/profile/Marc-Oliver_Loewner/publication/260713226_Aktuelle_Trends_in_der_Entwicklung_von_CityGML_3.0/links/0f317535f6e650c769000000.pdf - abgerufen am 20. April 2015.
- [Nissen et al. 2011] Nissen, Flemming; Friis-Christensen, Anders; Nielsen, Åge; Münster-Svensden, Jorgen; Bo Rykov, Jens: Framework for Specifying Transformation Rules. ESDIN Forschungsbericht D10.2 (ECP - 2007 - GEO - 317008, Version 1.0 Final, 2011. URL http://www.esdin.eu/sites/esdin.eu/files/ESDIN_D10_2_Framework_for_Trans_Rules_v1.0.pdf - abgerufen am 19. April 2015.
- [OGR GML 2015] OGR Geography Markup Language Model. URL http://www.gdal.org/drv_gml.html - abgerufen am 20. September 2015.
- [Reit 2013] Reitz, Thorsten: Approaches & Languages for Schema Transformation. INSPIRE KEN Workshop on Schema Transformation. Paris, Frankreich, 08. Oktober 2013. URL http://www.eurogeographics.org/sites/default/files/Schema%20Transformation%20Languages_T.Reitz_.pdf - abgerufen am 23. September 2015.
- [Reitz und Templer 2011] Reitz, Thorsten; Templer, Simon: HUMBOLDT Alignment Editor Manual. Version 2.1.0.M1, 28. Februar 2011. URL https://www.igd.fraunhofer.de/sites/default/files/hale_manual_en_2.1.0-M1-1.pdf - abgerufen am 22. April 2015.
- [Roschlaub 2013] Roschlaub, Robert: Aktuelles aus der PG 3D - Geobasisdaten. Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesre-

- publik Deutschland. Vortrag, 2013. URL http://www.3d-stadtmodelle.org/3d-stadtmodelle_2013/vortraege/03_Roschlaub_Aktuelles_aus_der_AdV.pdf - abgerufen am 03. September 2015.
- [Safe Inc. 2015] Safe Software Inc.: Homepage. URL <http://www.safe.com/> - abgerufen am 29. Mai 2015.
- [SIG 3D 2015] Special Interest Group 3D (SIG 3D): Homepage. URL <http://www.sig3d.org/> - abgerufen am 31. Juli 2015.
- [Spatialytics 2015] Spatialytics solutions Inc.: GeoKettle Projekt Homepage. URL <http://www.spatialytics.org/projects/geokettle/> - abgerufen am 29. Mai 2015.
- [Stachowiak 1973] Stachowiak, Herbert: Allgemeine Modelltheorie. Wien: Springer Verlag, 1973. URL https://archive.org/stream/Stachowiak1973AllgemeineModelltheorie/Stachowiak%20%281973%29%20Allgemeine%20Modelltheorie_djvu.txt - abgerufen am 19. Juli 2015.
- [Staub 2009] Staub, Peter: Über das Potenzial und die Grenzen der semantischen Interoperabilität von Geodaten. Zürich: Eidgenössische Technische Hochschule Zürich, Dissertation, 2009.
- [Unger 2012] Unger, Eva-Maria: Semantische Modelltransformation im Kontext von INSPIRE. Wien, Technische Universität Wien, Diplomarbeit, 2012.
- [Vautard 2013] Vautard, Edith: Schema Transformation of Administrative Data with GeoKettle. In: *INSPIRE KEN and EuroSDR workshop «Schema Transformation»*, Marne-la-Vallée, Frankreich, 8. - 9. Oktober 2013. URL http://www.eurogeographics.org/sites/default/files/GeoKettle_E.Vautard.pdf - abgerufen am 07. April 2015.
- [Wikipedia 2015a] Wikipedia: Die freie (online) Enzyklopädie. Artikel zu Geoinformatik. URL <https://de.wikipedia.org/wiki/Geoinformatik> - abgerufen am 23. Juni 2015.
- [Wikipedia 2015b] Wikipedia: Die freie (online) Enzyklopädie. Artikel zu World Wide Web Consortium. URL https://de.wikipedia.org/wiki/World_Wide_Web_Consortium - abgerufen am 03. August 2015.
- [Wikipedia 2015c] Wikipedia: Die freie (online) Enzyklopädie. Artikel zu Synergiepotential. URL <https://de.wikipedia.org/wiki/Synergiepotenzial> - abgerufen am 03. August 2015.
- [Wikipedia 2015d] Wikipedia: Die freie (online) Enzyklopädie. Artikel zu XPath. URL <https://de.wikipedia.org/wiki/XPath> - abgerufen am 03. August 2015.
- [Zedlitz 2013] Zedlitz, Jesper: Konzeptuelle Modellierung mit UML und OWL - Untersuchung der Gemeinsamkeiten und Unterschiede mit Hilfe von Modelltransformationen. Kiel: Christian-Albrechts-Universität Kiel, Dissertation, 2013.

Selbstständigkeitserklärung

Erklärung gemäß §12 Absatz 8 ADPO der Technischen Universität München:

„Ich versichere, dass ich diese Diplomarbeit / Master's Thesis selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe“

Sebastian Georg Kroiß

München, den 28.09.2015

Anhang A

Abkürzungsverzeichnis

| | |
|-----------------|--|
| AAA | AFIS - ALKIS - ATKIS |
| ADE | Application Domain Extension |
| Adv | Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland |
| AFIS | Amtliches Festpunktinformationssystem |
| ALKIS | Amtliches Liegenschaftskatasterinformationssystem |
| ATKIS | Amtliches Topographisch-Kartographisches Informationssystem |
| BayDSchG | Bayerisches Denkmalschutzgesetz |
| BLfD | Bayerisches Landesamt für Denkmalpflege |
| BU | Buildings |
| CityGML | City Geography Markup Language |
| CIM | Computation Independent Model |
| CSV | Comma-separated values |
| DS | Data Specification |
| EPSG | European Petroleum Survey Group Geodesy |
| ESDI | European Spatial Data Infrastructure |
| ESDIN | European Spatial Data Infrastructure with a Best Practice Network |
| EU | Europäische Union |
| ETL | Extract - Transform - Load |
| FME | Feature Manipulation Engine |
| GDI | Geodateninfrastruktur |
| GDI-BY | Geodateninfrastruktur Bayern |
| GDI-DE | Geodateninfrastruktur Deutschland |
| GeoZG | Geodatenzugangsgesetz des Bundes |
| GFM | General Feature Model |
| GNU | General Public License |
| GMES | Global Monitoring of Environment and Security |
| GML | Geography Markup Language |
| gOML | Geospatial Ontology Mapping Language |
| GUI | Graphical User Interface |
| HALE | Humboldt ALignment Editor |

IGD Fraunhofer Institut für Geographische Datenverarbeitung
ISO International Organization for Standardization
IUCN International Union for Conservation of Nature
INSPIRE INfrastructure for SPatial InfoRmation in Europe
JRC Joint Research Centre
KB KiloByte
LfU Bayerisches Landesamt für Umwelt
LoD Level of Detail
MDA Model-Driven Architecture
MOF Meta Object Facility
NK Normales Kriterium
OGC Open Geospatial Consortium
OK Optionales Kriterium
OMG Open Management Group
OML Ontology Mapping Language
OWL Web Ontology Language
PIM Platform Independent Model
PK Pflichtkriterium
PM Platform Model
PS Protected Sites
PSM Platform Specific Model
QVT Query/View/Transform
RIF Rule Interchange Format
SDI Spatial Data Infrastructure
UML Unified Modeling Language
UNESCO United Nations Education, Scientific and Cultural Organization
URI Uniform Resource Identifier
URN Uniform Resource Name
W3C World Wide Web Consortium
WFS Web Feature Service
XFMap XML FME Map
XML Extensible Markup Language
XPath XML Path Language
XSD Extensible Schema Definition
XSLT Extensible Stylesheet Language for Transformations

Anhang B

Digitaler Anhang (DVD)

Inhaltsverzeichnis

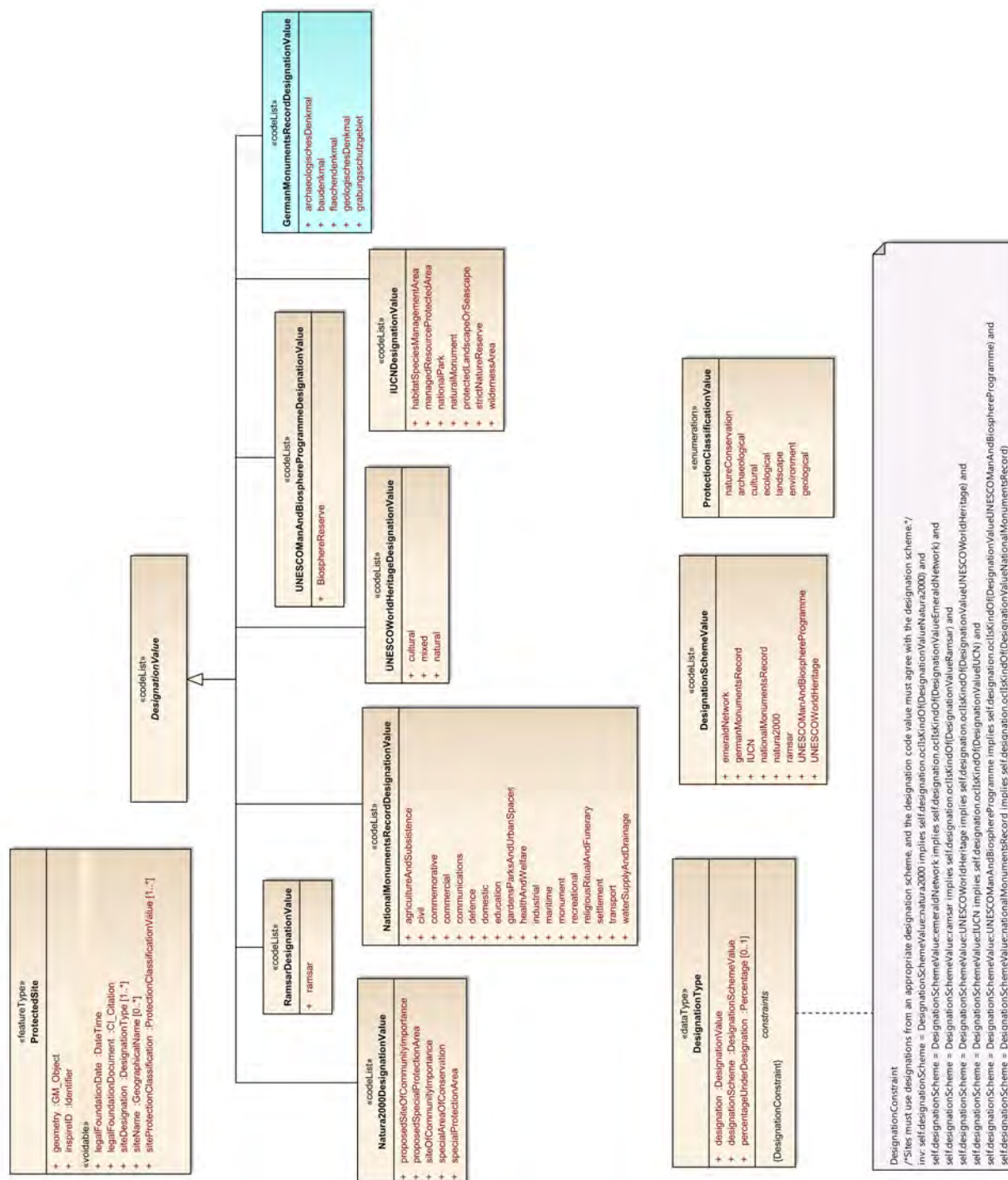
- Master's Thesis als PDF: *Thesis_Kroiss_Modelltrafowerkzeuge_20150928.pdf*
- Anhang E: *AnhangE_Bewertungsschema_Tabelle.pdf*
- Anhang F.1: *AnhangF1_Bewertungsschema_Diagramm_Teil_1.pdf*
- Anhang F.2: *AnhangF2_Bewertungsschema_Diagramm_Teil_2.pdf*
- Anhang F.1: *AnhangF1_Bewertungsschema_Diagramm_Teil_1_FME.pdf*
- Anhang F.2: *AnhangF2_Bewertungsschema_Diagramm_Teil_2_FME.pdf*
- Abbildungstabellen (Ordner)
 - Abbildungstabelle: INSPIRE Data Specification on Protected Sites - Denkmal BLfD (XSLX)
 - Abbildungstabelle: INSPIRE Data Specification on Protected Sites - Schutzgebiete LfU (XSLX)
 - Abbildungstabelle: INSPIRE Data Specification on Buildings - AAA 6.0 CityGML 1.0 (XSLX)
- Benötigte Dateien (Ordner)
- Daten (Ordner)
 - 01 FME Denkmaeler BLfD (Unterordner)
 - 02 FME Schutzgebiete LfU (Unterordner)
 - 03 HALE CityGML (Unterordner)
 - 04 HALE Denkmaeler BLfD (Unterordner)
 - 05 HALE Schutzgebiete LfU (Unterordner)
- Präsentationen (Ordner)
 - Zwischenpräsentation vom 06. Juli 2015 (PDF)
 - Abschlusspräsentation vom 23. September 2015 (PDF)
- Programmdateien (Ordner)
 - 01 FME-Dateien (Unterordner)
 - 02 HALE-Dateien (Unterordner)

Anhang C

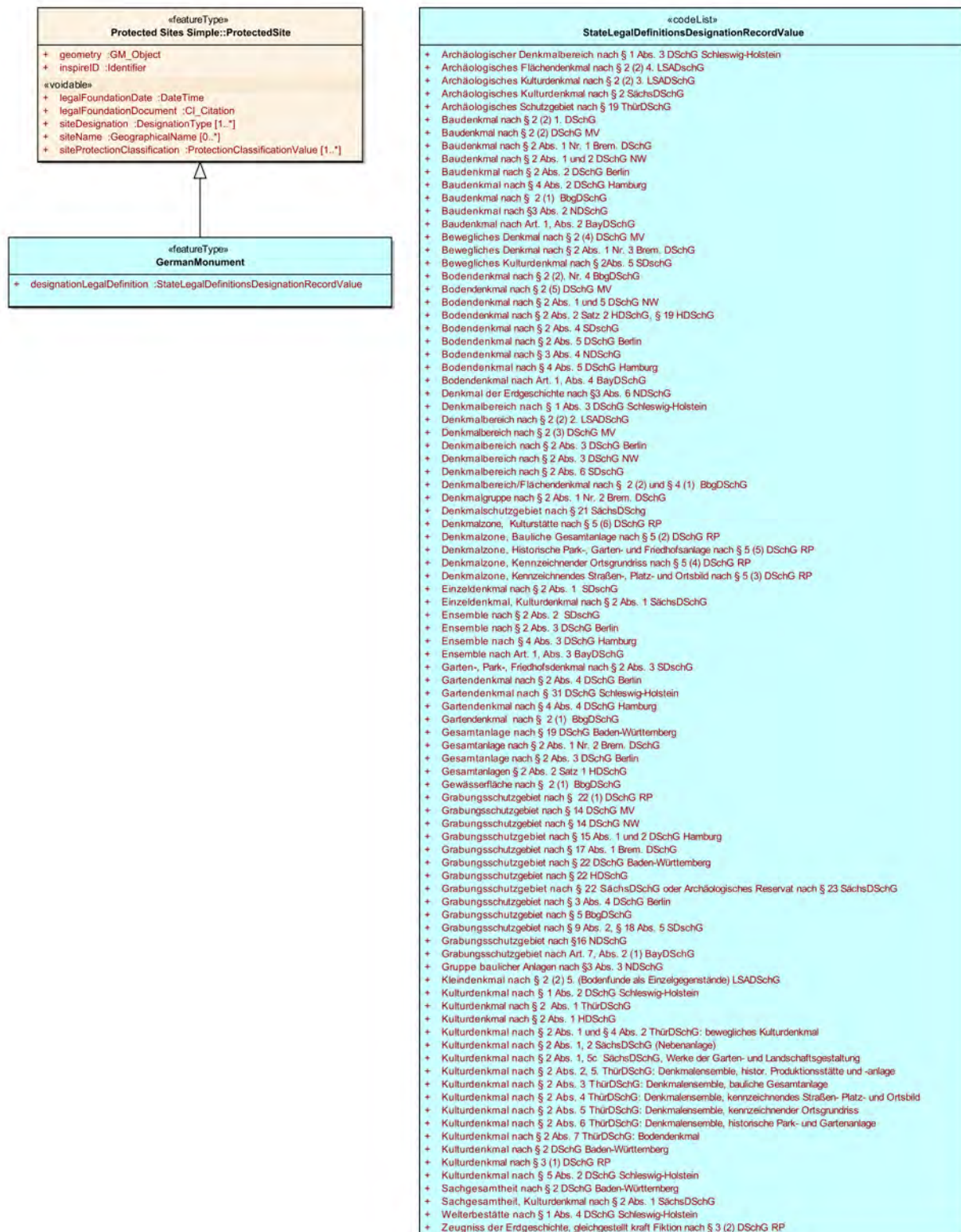
Klassendiagramme der INSPIRE Data Specifications

- **C.1 INSPIRE Data Specification on Protected Sites**
- **C.2 «featureType» GermanMonument mit zugehöriger «codeList»**
- **C.3 INSPIRE Data Specification on Buildings**

C.1 INSPIRE Data Specification on Protected Sites: Objektklasse und Codelists mit deutscher Erweiterung

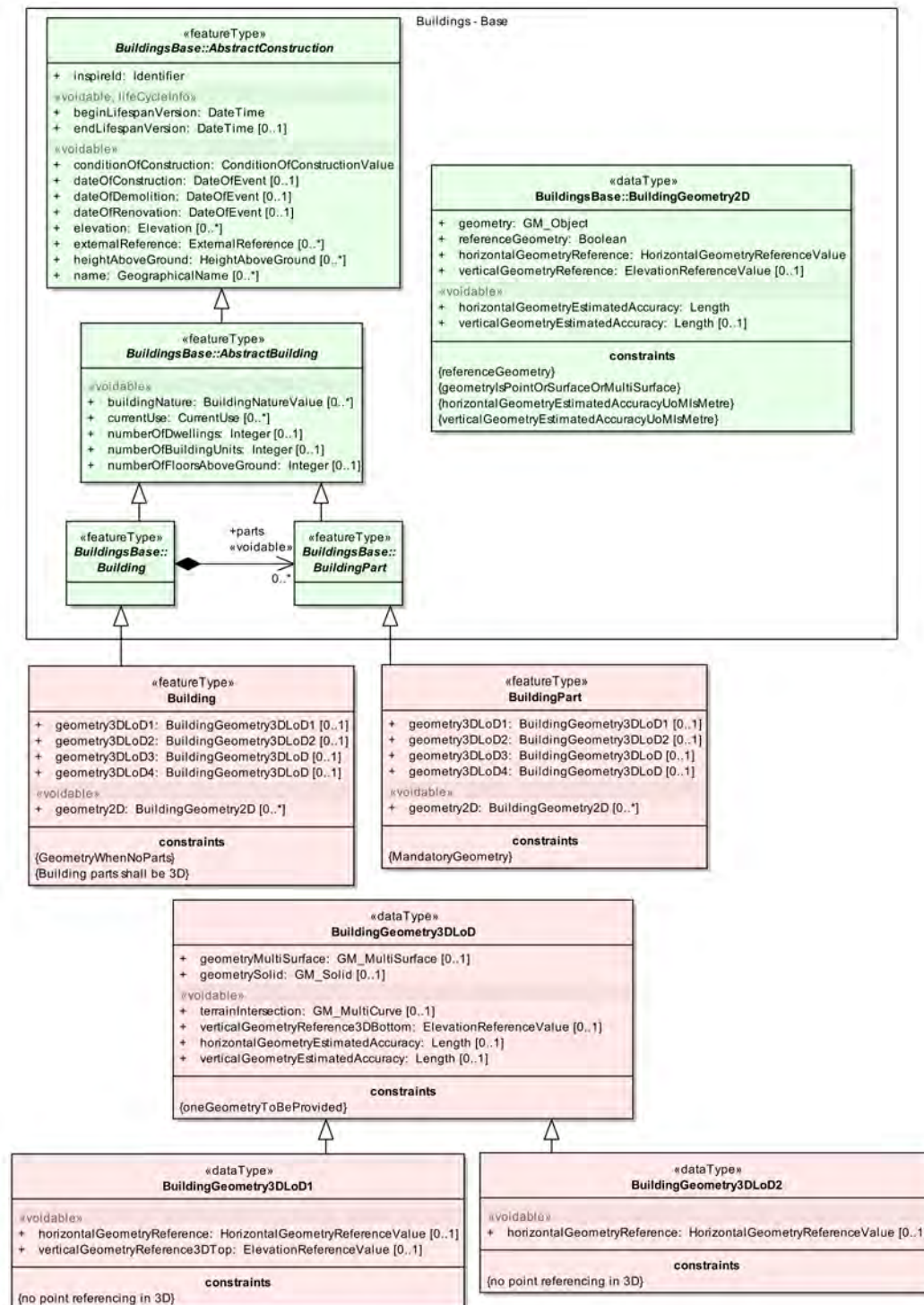


C.2 «featureType» GermanMonument und «codeList» StateLegalDefinitionDesignationRecordValue



C.3 INSPIRE Data Specification on Buildings: Buildings Core 3D

[INSPIRE DS BU 2013, S.59]



Anhang D

Klassifikation von Transformationsansätzen

- Unterscheidungsmerkmale zur Klassifikation von Transformationsansätzen nach Fichtinger (2011, S. 50)

