

INTERLIS 2 – GML 3

Eine Vergleichsstudie

Stephan Nebiker, Susanne Bleisch, Adrian Annen

Studienauftrag der KOGIS

KOGIS c/o Bundesamt für Landestopographie
Seftigenstrasse 264
CH-3084 Wabern

2. Juli 2004 (revidierte Fassung)

INHALTSVERZEICHNIS

Zusammenfassung	1
1. Einleitung	3
1.1. Ausgangslage	3
1.2. Zielsetzung der Studie	3
1.3. Gliederung des Berichts	4
2. Geonormen – eine Übersicht	6
2.1. Bedeutung von Geonormen	6
2.2. Arten von geo-relevanten Normen	6
2.3. Entwicklungen und Trends	8
3. Basiskonzepte und -technologien	10
3.1. Einleitung und Übersicht	10
3.2. Die OMG-Standards rund um UML	10
3.2.1. UML	11
3.2.2. MOF	11
3.2.3. XMI	12
3.2.4. MDA – Model Driven Architecture	12
3.3. Die W3C-Standards rund um XML	13
3.3.1. XML	14
3.3.2. XML Schema	14
3.3.3. XLink	14
3.3.4. XSL und XSLT	15
3.3.5. Schematron	15
4. Vergleich von INTERLIS 2 und GML 3	19
4.1. Standardisierung – Gremien, Prozesse und Produkte	19
4.2. Grundprinzipien	21
4.2.1. Modellierungskonzept	21
4.2.2. Objekt-Konzept	21
4.3. Ausgewählte Aspekte beider Standards	23
4.3.1. Beschreibungssprache	23
4.3.2. Assoziationen / Beziehungen	25
4.3.3. Geometrie und Topologie	25
4.3.4. Einheiten und Bezugssysteme	27
4.3.5. Konsistenzbedingungen	28

4.4.	Transfer	30
4.4.1.	Domänenspezifische Datenmodelle bzw. Applikationsschemas	30
4.5.	Neuerungen in GML 3.1	31
5.	Praktische Untersuchungen	33
5.1.	Modellierung eines INTERLIS-Themas in GML 3	34
5.1.1.	Untersuchter Ausschnitt aus dem INTERLIS 2-Testdatensatz	34
5.1.2.	Entsprechendes Datenschema in GML 3	35
5.2.	Abbildung zwischen INTERLIS 2 und GML 3	37
5.2.1.	Abbildungsregeln	37
5.2.2.	Abbildungsprobleme	38
5.3.	Probleme im INTERLIS-Testdatensatz	40
5.4.	Konvertierungsvarianten	40
5.4.1.	Variante 1 – Direkte Datenkonvertierung	40
5.4.2.	Variante 2 – modell-basierte Transformation zwischen INTERLIS XML-Schema und GML-Applikationsschema	41
5.4.3.	Variante 3 – modell-basierte Transformation zwischen den normativen Modellbeschreibungen	42
5.4.4.	Variante 4 – modell-basierte Transformation mittels UML	43
5.4.5.	Zusammenstellung und Vergleich der Varianten	44
6.	Zusammenfassung und Beurteilung	45
6.1.	INTERLIS 2 – Stärken und Schwächen	45
6.2.	GML 3 – Stärken und Schwächen	46
6.3.	Wichtigste Gemeinsamkeiten	47
6.4.	Wichtigste Unterschiede	48
6.5.	Fazit	48
7.	Weiteres Vorgehen – Szenarien und Empfehlungen	49
7.1.	Szenario 1 – Unabhängige Weiterentwicklung von INTERLIS	49
7.2.	Szenario 2 – Zweigleisige Strategie mit Konvertierungsschiene	50
7.3.	Szenario 3 – Rasche Umstellung auf GML	51
7.4.	Empfehlungen und Ausblick	51
8.	Literatur / Referenzen	53
8.1.	Literatur	53
8.2.	Online-Ressourcen	53
9.	Anhang.....	1
9.1.	Anhang 1 – Typzuordnung zwischen INTERLIS 2 und GML 3	1
9.2.	Anhang 2 – Beispiel AREA in GML	3

Zusammenfassung

Im Rahmen dieser Studie werden mit INTERLIS 2 und GML 3 zwei moderne Mechanismen für die Beschreibung und den Austausch von Geodaten analysiert und miteinander verglichen. Beide sind in ihrer Zielsetzung sehr ähnlich, sie unterscheiden sich jedoch deutlich in ihrem Aufbau und in ihrer Entwicklungsgeschichte. Mit der Studie sollen die speziellen Eigenschaften, Stärken und Schwächen sowie das (Synergie-) Potential der beiden Mechanismen aufgezeigt werden.

Dazu erfolgen eingangs eine Analyse und Dokumentation der verwendeten Basiskonzepte und -technologien. Diese werden gefolgt von einer Gegenüberstellung der wichtigsten Aspekte beider Standards. Die Ergebnisse einer Reihe von praktischen Untersuchungen zeigen anschliessend Möglichkeiten einer Interoperabilität zwischen beiden Mechanismen auf, welche einen Übergang von einem Standard zum andern oder einen kombinierten Einsatz ermöglichen würden. Nach diesen technischen Untersuchungen werden verschiedene mögliche Szenarien für die Weiterentwicklung und den Einsatz der beiden Geonormen im Schweizerischen Kontext aufgezeigt.

Die Resultate der Untersuchungen können wie folgt zusammengefasst werden:

- INTERLIS und GML sind grundsätzlich gleichwertig, in vielen Belangen aber auch andersartig. Beide Mechanismen weisen eine objekt-orientierte Basis auf und ermöglichen einen modell-basierten Datenaustausch.
- Die GeoLanguage INTERLIS könnte man als 'klein aber fein' charakterisieren. Sie ist elegant, bewährt und stabil. INTERLIS weist eine strikte Trennung auf zwischen der Modellbeschreibung und der Beschreibung des Transfers. INTERLIS ist eine eigenständige, gut verständliche Sprache, erfordert aber speziell entwickelte Softwaretools. Die regionale Verbreitung von INTERLIS ist begrenzt und konzentriert sich auf die Schweiz und auf wenige weitere Staaten.
- Die Geography Markup Language GML könnte man mit den Attributen 'gross, dynamisch und komplex' versehen. GML zeichnet sich durch einen konsequenten Einsatz von XML aus. Dies ermöglicht den Einsatz standardisierter Softwaretools, erfordert aber gewisse Kompromisse bei der Datenmodellierung und bei der Lesbarkeit der Modellbeschreibungen. Der Normierungsprozess von GML ist stark im Fluss mit vielen Versionen in kurzer Zeit, was dazu führt, dass die Umsetzung stark hinterher hinkt. Es ist zu hoffen, dass die laufende ISO-Normierung hier zu einer gewissen Konsolidierung führen wird.
- Die praktischen Untersuchungen zeigten, dass sowohl Modelltransformationen als auch die Konvertierung der Daten zwischen INTERLIS und GML machbar sind. Bei der Modelltransformation stellte sich die Abbildung zwischen den beiden normativen Modellbeschreibungen in INTERLIS und GML als interessanter und gangbarer Weg heraus. Die Untersuchungen ergaben, dass eine modell-basierte Datenkonvertierung mit XSL möglich ist, und dass auch die dazu erforderlichen Abbildungsvorschriften (XSL Style Sheets) automatisch abgeleitet werden können.

- Für die praktische Umsetzung einer Interoperabilität zwischen INTERLIS und GML wird die Erweiterung des INTERLIS-Compilers um einen GML-Output-Generator vorgeschlagen. Dieser GML-Generator sollte einerseits das GML-Applikationsschema für das jeweilige INTERLIS-Datenmodell erzeugen und andererseits das XSL Style Sheet generieren. Damit könnten mit jedem beliebigen XSL-Prozessor alle entsprechenden INTERLIS-Datensätze konvertiert werden. Diese technische Lösung würde zudem einen kombinierten Einsatz von INTERLIS und GML ermöglichen – mit INTERLIS als weitgehend konkurrenzloser Datenmodellierungssprache und GML als Transfermechanismus.
- Die Studie schliesst ab mit dem Skizzieren von drei möglichen Szenarien für die zukünftige Strategie in der schweizerischen Geonormierung. In der favorisierten, zweigleisigen Strategie wird die bestehende INTERLIS-Modellierung um einen GML-Transfer erweitert (vgl. oben). Dieses Szenario ermöglicht eine rasche Unterstützung der GML-Technologie auf der bewährten INTERLIS-Modellierungsbasis bei vergleichsweise geringem Risiko. Damit könnten das exzellente CH-Know-how im modell-basierten Geodaten austausch und der reiche Fundus an domänenspezifischen Datenmodellen in das GML-Umfeld eingebracht werden.

1. Einleitung

1.1. Ausgangslage

Mit der 'GeoLanguage' INTERLIS existiert seit über 10 Jahren ein bewährter und gleichzeitig moderner Mechanismus für die Beschreibung und den Austausch von Geodaten. In der Schweiz ist INTERLIS Version 1 (INTERLIS 1) in der Gesetzgebung über die amtliche Vermessung verankert und in einer nationalen Norm verbindlich definiert. Mit der soeben abgeschlossenen Entwicklung von INTERLIS 2 werden neue Informatikkonzepte sowie eine deutlich erweiterte Funktionalität unterstützt. INTERLIS wurde von Beginn weg für ein mehrsprachiges Umfeld konzipiert und ist daher für einen internationalen Einsatz prädestiniert. Zurzeit beschränkt sich jedoch die geographische Verbreitung auf die Schweiz und auf Staaten mit einer Entwicklungszusammenarbeit mit der Schweiz.

Im Jahre 2000 publizierte das OpenGIS Consortium (OGC) mit Version 1 der Geography Markup Language (GML 1) einen Mechanismus mit einer vergleichbaren Zielsetzung wie derjenigen von INTERLIS. OGC ist ein internationales Konsortium mit Mitgliedern aus Industrie, Verwaltung und Hochschulen, welches sich die Schaffung von offenen, weltweit gültigen Standards für den Umgang mit Geodaten zum Ziel gesetzt hat. Seit der Publikation von GML 1 wurde die Spezifikation bereits mehrmals erweitert und revidiert und liegt zurzeit als Version 3.0 vor. Die meisten Hersteller konnten mit diesem Entwicklungstempo bisher nicht Schritt halten und unterstützen – falls überhaupt – bestenfalls GML 2. Auf Grund der mittlerweile sehr starken Stellung des OpenGIS Consortiums werden jedoch die meisten GI-Softwarehersteller nicht umhin kommen, dessen wichtigste Spezifikationen, und damit auch GML, in ihren Produkten umfassend zu unterstützen.

Aus dieser Konstellation ergibt sich für die Entwickler und Anwender von INTERLIS eine Reihe von Fragen, wie zum Beispiel:

- Inwiefern sind GML und INTERLIS tatsächlich vergleichbar?
- Welche Konsequenzen hätte ein internationaler Durchbruch von GML auf die Weiterentwicklung und den Einsatz von INTERLIS?
- Welche möglichen Strategien betreffend zukünftiger Unterstützung von INTERLIS und GML bieten sich Koordinationsstellen, Softwareherstellern und Geodatenproduzenten allenfalls an?

Vor diesem Hintergrund erhielt die Fachhochschule beider Basel (FHBB) von der Koordinationsstelle des Bundes für geografische Information und geografische Informationssysteme (KOGIS) den Auftrag zur Durchführung der vorliegenden Vergleichsstudie zwischen INTERLIS 2 und GML 3.

1.2. Zielsetzung der Studie

Gegenstand dieses Studienauftrags war die Erarbeitung einer "Gegenüberstellung von INTERLIS 2 und GML 3, um eine nachvollziehbare, aussagekräftige und transparente

Analyse über die spezifische Eignung und die Einsatzmöglichkeiten der beiden 'Produkte' zu erhalten". Dazu sollten Stärken und Schwächen der beiden Mechanismen eruiert und verglichen werden.

Insbesondere sollte diese Studie Aussagen ermöglichen, in welche Richtung INTERLIS in Zukunft weiter entwickelt werden soll. Dabei reichen die denkbaren Szenarien von einer Weiterentwicklung unabhängig von GML, über einen zweigleisigen Ansatz mit der Entwicklung verschiedener Konvertierungstools für Schemas und Daten bis hin zu einer vollständigen Umstellung auf GML.

Da die nächste GML-Version (GML 4.0) eine ISO-Norm werden soll, hat dies zur Folge, dass diese Norm via CEN auch für die Schweiz gelten wird. Diese Studie soll aufzeigen, welche Vorkehrungen die Schweiz treffen könnte und sollte, um auf diese Norm am besten gewappnet zu sein.

Die Studie soll sich dazu sowohl auf theoretische Überlegungen als auch auf praktische Untersuchungen abstützen.

Demgegenüber soll und kann die Studie die folgenden Punkte **nicht** abdecken:

- Eine vertiefte Einführung in INTERLIS oder GML
- Eine vertiefte Einführung in objektorientierte Konzepte und in die objektorientierte Modellierung.
- Eine vertiefte Einführung in die dabei zum Einsatz kommenden Basistechnologien und –standards, wie UML, XMI, XML, XML Schema, Schematron etc.

Die angesprochenen Konzepte und Standards werden daher nur kurz eingeführt und positioniert und mit Verweisen auf entsprechende Literaturstellen und Online-Ressourcen versehen.

1.3. Gliederung des Berichts

Der vorliegende Bericht besteht aus einem einleitenden Teil (Kapitel 1 bis 3), aus den eigentlichen Untersuchungen (Kapitel 4 und 5) sowie der Beurteilung und den Empfehlungen (Kapitel 6 und 7). Die einzelnen Kapitel haben die folgenden Schwerpunkte:

Kapitel 2 – Dieses Kapitel bietet eine kurze Übersicht über die Bedeutung von Geonormen, über verschiedene Typen von Geonormen und über aktuelle Entwicklungen und Trends in diesem Bereich.

Kapitel 3 – In diesem Kapitel werden zwei wichtige Familien von Standards aus der Informatik bzw. Informationstechnologie vorgestellt, welche die Basis beider untersuchter Mechanismen bilden. Zum einen sind dies die OMG-Standards im Umfeld von UML der OMG, zum andern die W3C-Standards rund um XML.

Kapitel 4 – Dieses Kapitel enthält eine detaillierte Gegenüberstellung der unterschiedlichen Aspekte von INTERLIS und GML, angefangen vom Standardisierungsprozess und den Grundprinzipien bis hin zur Umsetzung in der Praxis.

Kapitel 5 – Dieser Abschnitt des Berichts enthält die Resultate einer Reihe von ergänzenden praktischen Untersuchungen, die im Rahmen der Diplomarbeit 2003 von Adrian Annen an der FHBB durchgeführt und anschliessend vertieft wurden.

Kapitel 6 – In diesem Kapitel werden die Stärken und Schwächen, sowie die wichtigsten Gemeinsamkeiten und Unterschiede von INTERLIS und GML zusammengestellt und beurteilt.

Kapitel 7 – Dieses Kapitel beschreibt mögliche Szenarien für das weitere Vorgehen mit den entsprechenden Chancen und Risiken und liefert erste Empfehlungen.

2. Geonormen – eine Übersicht

In diesem Abschnitt wird der wichtigste Nutzen von Geonormen anhand einiger Kriterien zusammengestellt. Zudem wird eine Taxonomie von Geonormen vorgeschlagen, die eine Positionierung der untersuchten Mechanismen ermöglichen soll.

2.1. Bedeutung von Geonormen

Der Term Geonormen wird hier als Oberbegriff für de jure und de facto Standards für die Modellierung, den Austausch und die Nutzung von Geodaten verwendet. Geonormen gewinnen mit der rasch zunehmenden Verbreitung und Vernetzung von Geoinformationssystemen und mit dem Vordringen in laufend neue Anwendungsgebiete schnell an Bedeutung.

Die Bedeutung und der praktische Nutzen von Interoperabilität und Normung für GIS-Anwender in der Schweiz (und anderswo) wird im gleichnamigen Bericht der SOGI-Fachgruppe GIS-Technologie dargelegt (SOGI Fachgruppe GIS-Technologie, 2003). Der Bericht führt als wichtigsten Nutzen für die AnwenderInnen die folgenden Punkte auf:

- Produktivitätssteigerung von Unternehmen und Behörden
 - durch die Nutzung genormter Programmschnittstellen
 - durch die Nutzung genormter Austauschformate und Mechanismen
- Nutzen durch Software spezifischer Technologien ab der Stange.
- Investitionsschutz (durch Plattformunabhängigkeit).
- Bessere Kontrolle der Daten.
- Klar definierte Richtlinien für Ausschreibungen durch die zwingende Vorgabe des abzugebenden Datenformats, und damit Chancengleichheit für alle Anbieter.
- Normung wird auf einer systemneutralen Daten- und Modellierungsebene geführt.
- Alle Beteiligten sprechen die gleiche "Sprache".

2.2. Arten von geo-relevanten Normen

Der Begriff geo-relevante Normen soll hier für die Gesamtheit von applikationsunabhängigen IT-Standards und von spezifischen Geoinformations-Standards verwendet werden, welche für die Konzeption, die Implementierung und den Betrieb von verteilten, zumeist webbasierten Geoinformationssystemen von Bedeutung sind.

Moderne Geoinformationssysteme sind ohne eine ganze Reihe von applikations-unabhängigen **IT-Normen** nicht mehr denkbar. Zu den bekanntesten Beispielen gehören:

- Datenbankstandards, wie SQL, ODBC und JDBC
- Internet- und Webstandards, wie TCP/IP, HTTP, HTML, XML

- Web Service Standards, wie SOAP, WSDL und UDDI
- so wie beispielsweise die Datenmodellierungssprache UML

So wie die IT-Standards decken auch die eigentlichen **Geonormen** ein immer breiteres Spektrum ab und ein Überblick wird zunehmend schwierig. Die meisten neuen Geonormen bauen auf existierenden IT-Standards auf, erweitern und kombinieren diese.

Taxonomie von Geonormen

Zurzeit existiert keine umfassende und breit akzeptierte Taxonomie von Geonormen – mit Ausnahme des Bereichs der Services (vgl. unten). Die folgende Momentaufnahme soll trotzdem einen Überblick über Positionierung und Zielsetzung einiger wichtiger Normierungsbereiche geben.

Nach unserer eigenen Taxonomie können Geonormen in eine oder mehrere der folgenden Klassen eingeteilt werden:

- Datencodierung und –formate
- Datenmodellierung
- Metadaten
- GI-Dienste / Services

Die Reihenfolge in dieser Aufstellung korrespondiert in etwa mit der chronologischen Reihenfolge in welcher die unterschiedlichen Typen von Geonormen entstanden. INTERLIS und GML decken dabei primär die Bereiche Geodatenmodellierung und Datencodierung ab. Als Modellierungs- und Codierungs-Normen sind sie aber auch wichtiger Bestandteil von darauf aufbauenden Normen, beispielsweise in den Bereichen Services.

Services

Wie eingangs erwähnt, existiert für den Bereich der Geoinformationsdienste (Services) eine publizierte Taxonomie. Das aus sechs Dienstklassen bestehende OGC Service Model (Open GIS Consortium & ISO/TC211, 2002), basiert beispielsweise auf der ISO-Norm 19101, welche das Extended Open Systems Environment (EOSE) Model for Geographic Information festlegt. Die sechs Dienstklassen des OGC Service Models sind (Bernard und Streit, 2002):

- Geographic Human Interaction Services – sind Dienste zur Realisierung von Nutzerschnittstellen, Editoren und Präsentationen. Beispiele sind Clients zur Kartenansicht, zur Anfrage an einen Katalogdienst oder zur Zusammenstellung von Dienstketten.
- Geographic Model/Information Services – dienen der Erfassung, Verwaltung und Fortführung von Geodaten, Metadaten und deren konzeptionellen Schemata. (Bsp. OGC Web Catalog Services, Web Feature Services, Web Coverage Services, Web Mapping Services)

- Geographic Workflow/Task Services – unterstützen die Anwender bei der Durchführung fachspezifischer Aufgaben, zu deren Lösung die Durchführung mehrerer aufeinander folgende Arbeitsschritte unter Einbindung unterschiedlicher Ressourcen erforderlich ist. (Bsp. Chain Definition Services und Workflow Enactment Services.)
- Geographic Processing Services – realisieren Algorithmen zur Bearbeitung von Geodaten. (Bsp. Web Coordinate Transformation Services, Web Gazetteer Services)
- Geographic Communication Services – bilden die Basis für den Austausch von Geodaten zwischen GI-Diensten. Beispiele sind Dienste zur Kodierung und Dekodierung von Geodaten oder zur Komprimierung von Geodaten für den Transfer. (Bsp. Geographic Markup Language GML)

2.3. Entwicklungen und Trends

Die Normierung in der Geoinformatik widerspiegelt die teilweise radikalen Veränderungen, welche der Fachbereich selbst in den letzten 10-15 Jahren durchlaufen hat. Diese Entwicklung ist bei weitem noch nicht abgeschlossen und wird sich auch in der Zukunft fortsetzen. Diesen Entwicklungen und Trends sollten zukünftige Standardisierungsbestrebungen Rechnung tragen, oder diese sogar vorwegnehmen, um auf lange Sicht nutzbar zu bleiben.

Einige der wichtigsten Entwicklungen und Trends im Bereich der Geonormen lassen sich wie folgt charakterisieren:

- **von Formaten zu Modellen** – Die Schwächen format-basierter Standards, namentlich deren fehlende Flexibilität und Erweiterbarkeit sowie die ungenügende Unterstützung für semantische Informationen, sind seit langem bekannt. Wegen ihrer Einfachheit und grossen Verbreitung, sind diese aber kaum unterzukriegen; dies obwohl seit Anfang der 90er-Jahre funktionierende modell-basierte Standards, wie INTERLIS, zur Verfügung stehen und sich in der Praxis bewähren. Es darf jedoch festgestellt werden, dass die Überlegenheit des modell-basierten Ansatzes mittlerweile erkannt ist und dass diese Erkenntnis in die laufenden internationalen Normierungsaktivitäten eingeflossen ist.
- **von Implementierungsvorschriften zu Schnittstellenspezifikationen** – Mit der zunehmenden Verteilung und Vernetzung von Geoinformationssystemen gewinnt die Spezifikation der Schnittstellen von (web-basierten) GI-Diensten rasch an Bedeutung. Dies ergibt sich aus der Erkenntnis, dass für die Realisierung interoperabler Dienste nicht deren Implementierung, sondern deren Methoden und Schnittstellen zu normieren sind. Allerdings sind Schnittstellenspezifikationen in der Regel nicht Ersatz sondern eine Ergänzung von Implementierungsvorschriften. So beruht beispielsweise der OGC Web Feature Service (WFS) (Open GIS Consortium, 2003b) auf einem Austausch von Geodaten, welche in GML codiert sind.
- **von der Struktur zum Verhalten** – Bei der Entwicklung von Geonormen stand die Modellierung der Struktur von Geodaten lange im Vordergrund. Dies gilt sowohl für format- als auch modell-basierte Mechanismen (z.B. DXF, Shapefiles, INTERLIS etc.).

Diesen strukturellen Modellen liegt eine Systemsicht zu Grunde, welche die Struktur von Objekten in den Vordergrund stellt, also deren Klassifizierung, deren Beziehungen und Attribute, sowie allenfalls deren Methoden. Bei einem Einsatz von UML werden diese Aspekte in aller Regel durch statische Klassendiagramme dargestellt. Im Gegensatz dazu stehen bei einer Modellierung des Verhaltens die Interaktionen und Kollaborationen zwischen Objekten im Vordergrund, welche wiederum für die Realisierung von Diensten von zentraler Bedeutung sind. In UML können diese mittels Zustands- und Aktivitätsdiagrammen modelliert werden.

- **von text-orientierter zu grafik-orientierter Modellierung und zurück** – Bei der Beschreibung von Geodaten spielte die Text-Repräsentation schon immer eine wichtige Rolle. Sie werden aber zusehends ergänzt durch graphische Repräsentationen der Datenmodelle. Die Datenbeschreibung in Textform, wie sie auch in INTERLIS zum Einsatz kommt, hat den Vorteil der formalen Exaktheit und der guten Lesbarkeit, insbesondere auch durch Programme. Zu den Nachteilen gehören die Notwendigkeit des Erlernens einer eigentlichen Sprache und die schwierige Kommunikation gewisser Aspekte, beispielsweise von Beziehungen und Vererbung. Graphische Beschreibungen in der Form der Unified Modeling Language (UML) – oder in der traditionelleren Form der Entitäts-Beziehungs-Diagramme (ERM) – erleichtern die Illustration komplexer Strukturen in einer relativ leicht verständlichen Weise. Ein grosser Nachteil von graphischen Repräsentationen wie UML war jedoch ihr 'informeller' Charakter bzw. das Fehlen einer formal exakten Modellbeschreibung. So können UML-Diagramme nur über eine zusätzliche Beschreibung zum Beispiel mit XMI (vgl. 3.2.3) zwischen verschiedenen Systemen ausgetauscht werden. Mit der Etablierung solcher Kombinationen von text- und grafik-orientierten Modellbeschreibungen lassen sich die Vorteile beider Repräsentationen nutzen. Der Trend in Richtung Round-Trip-Modellierung, welche eine wechselseitige graphische und textuelle Modellbearbeitung und –darstellung zulässt, ist im INTERLIS/UML-Editor bereits ansatzweise realisiert. Allerdings ist dort die graphische Darstellung die "Masteransicht", woraus die textuelle Darstellung in INTERLIS abgeleitet werden kann. Umgekehrt kann eine INTERLIS-Beschreibung importiert werden, wobei die resultierende Grafik manuell editiert werden muss. Die wechselweise Bearbeitung von grafischer und textueller Repräsentation (wie bei einer echten Round-Trip-Modellierung) ist zur Zeit noch nicht möglich.

3. Basiskonzepte und -technologien

3.1. Einleitung und Übersicht

INTERLIS und GML sind keine isolierten Standards, sondern sie stützen sich auf verschiedene Basis-IT-Standards ab. Die wichtigsten internationalen Standards zur Beschreibung und Codierung von Daten können dabei in die folgenden zwei Familien unterteilt werden:

- OMG-Standards rund um UML (UML, MOF, XMI, MDA)
- W3C-Standards rund um XML (XML, XML Schema, XLink, XSLT etc.)

Sowohl INTERLIS als auch GML stützen sich stark auf Standards aus diesen beiden Familien ab. Aus diesem Grund werden in diesem Abschnitt die wichtigsten Basisstandards kurz vorgestellt und positioniert; vorgängig zum eigentlichen Vergleich von INTERLIS und GML im nächsten Abschnitt.

3.2. Die OMG-Standards rund um UML

Die Object Management Group (OMG) ist ein offenes, non-profit Industrie-Konsortium, welches Informatik-Standards für interoperable Anwendungen entwickelt und wartet. Zu den OMG-Mitgliedern zählen fast alle grossen Softwarefirmen, sowie Hunderte von kleineren Unternehmen. Zu den bekanntesten OMG-Standards gehören die Middleware-Architektur CORBA und die objektorientierte Modellierungssprache UML. Die laufenden Normierungsaktivitäten der OMG konzentrieren sich auf die Spezifikation der Model Driven Architecture (MDA), welche die (automatisierte) Entwicklung plattform-spezifischer Anwendungen auf der Basis plattform-unabhängiger Modelle anstrebt. OMG-Spezifikationen können von der OMG-Website (www.omg.org) kostenlos heruntergeladen werden.

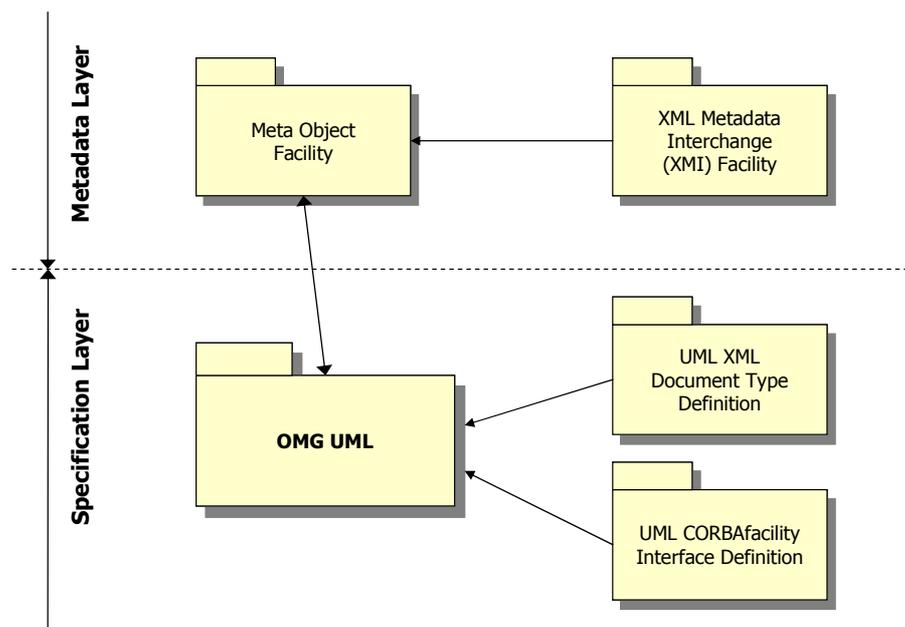


Abbildung 1: Beziehungen zwischen den verschiedenen OMG-Technologien

3.2.1. UML

Die Unified Modeling Language (UML) ist eine Sprache, um Artefakte eines objekt-orientierten Softwaresystems zu spezifizieren, visualisieren, konstruieren und dokumentieren. Als Artefakt gilt hier Information, die in einem Software-Entwicklungsprozess benutzt oder durch ihn produziert wird und eine gewisse Dauerhaftigkeit aufweist, dies im Gegensatz zu flüchtiger Information, wie gesprochene Worte oder Gesten. UML verfügt über eine Notation und ein Metamodell. Die Notation ist die Syntax der Modellierungssprache, sie definiert grafische Elemente und Konzepte zur Modellierung (beispielsweise Klasse, Generalisierung, Assoziation etc.). Das Metamodell dient dazu, diese Elemente methodisch und syntaktisch konsistent zu beschreiben und exakt in Gestalt von Diagrammen darzustellen. (OMG Object Management Group, 2003) und (Andresen, 2003)

UML darf wohl zu Recht als DIE Modellierungssprache schlechthin für objektorientierte Softwaresysteme bezeichnet werden. Sie bewährt sich für die Modellierung grosser, komplexer Systeme und wird heute durch eine Reihe mächtiger Softwarewerkzeuge unterstützt.

Komponenten von UML

Die UML standardisiert die Darstellung der Analyse und Designs objektorientierter Systeme. Als graphische Modellierungssprache besteht sie aus einem Dutzend Diagrammtypen, angefangen von 'Use Case'- und Aktivitätsdiagrammen für die Anforderungsanalyse und Verhaltensmodellierung, über Klassendiagramme für die strukturelle Modellierung bis hin zu Komponenten- und Subsystem-Diagrammen für die Spezifikation der Implementierung und Verteilung von Systemkomponenten.

UML in der Geodatenmodellierung

Bei der Beschreibung und dem Austausch von Geodaten standen bisher Datenstrukturen und allenfalls Schnittstellen im Vordergrund. Für die Repräsentation dieser 'statischen' Sicht werden primär **Klassendiagramme** und **Paketdiagramme** eingesetzt. Mit dem Trend in Richtung Geoinformationsdienste und komplexe Geodateninfrastrukturen gewinnen auch andere Diagrammtypen wie Sequenzdiagramme, Zustandsdiagramme und Verteilungsdiagramme zunehmend an Bedeutung.

3.2.2. MOF

Im Vergleich zu UML ist die MetaObject Facility (MOF) der OMG (OMG Object Management Group, 2002) in der Regel deutlich weniger bekannt, dies obwohl sie unter anderem das Fundament von UML bildet. MOF stellt ein standardisiertes Metamodell für objektorientierte Analyse und Entwurf zur Verfügung. Da UML-Modelle auf dem MOF Metamodell basieren, können diese unter Zuhilfenahme von XMI (vgl. unten) zwischen UML-Softwaretools frei ausgetauscht werden. Ohne ein solch gemeinsames Metamodell wäre dies nicht praktikabel.

Hinweis: MOF kann als Metamodell für beliebige objekt-orientierte Systeme dienen. MOF

ist somit nicht 'nur' das Metamodell von UML, auch wenn UML klar das wichtigste darauf basierende Metamodell ist.

3.2.3. XMI

XML Metadata Interchange (XMI) ist eine OMG-Spezifikation, welche den Austausch von Datenmodellen erlaubt, welche auf MOF aufbauen. Damit können beispielsweise auch UML-Modelle systemneutral zwischen verschiedenen Modellierungswerkzeugen ausgetauscht werden.

XMI definiert Abbildungsregeln zwischen MOF und XML. Im Falle von UML wird die XML-Abbildung durch die XMI DTD für UML bzw. durch das XML Schema für UML festgelegt. Diese UML-Abbildung ist denn auch das bekannteste XMI Schema.

Hinweis: XMI ist nicht nur eine spezifische DTD bzw. ein XML Schema für den Austausch von UML-Modellen! Es können damit auch andere MOF-kompatible Datenschemas ausgetauscht werden.

XMIs Reverse Mapping

Mit XMI können grundsätzlich beliebige XML Schemas, also auch Schemas, welche nicht einem XMI XML Schema entsprechen, auf ein MOF Metamodell abgebildet werden. Allerdings kann ein allgemeines XML Schema, welches nicht einem XMI XML Schema entspricht, nicht die gesamte Semantik eines MOF-Metamodells ausdrücken (Frankel, 2003). Da aber die XMI-Abbildung parametrisierbar ist, können die fehlenden Informationen über Parameter vorgegeben werden, welche beispielsweise festlegen, um welchen Beziehungstyp (Komposition oder Aggregation) es sich im speziellen Fall handeln soll.

3.2.4. MDA – Model Driven Architecture

Bei der Entwicklung der Model Driven Architecture MDA® (OMG Object Management Group, 2004) durch die OMG dürfte der Leitspruch Pate gestanden haben, dass 'Veränderung das einzig Konstante im Leben ist' oder zumindest, dass Änderungen auch in der Informatik normal sind. Eine Folge dieser Veränderungen ist die zunehmende Forderung nach Interoperabilität zwischen unterschiedlichen Plattformen, Anwendungen und Unternehmungen. Dieser Trend hat sich mit der raschen 'Ausbreitung' des Internets in sämtliche Aspekte von Unternehmungen und mit laufend neuen Technologien und Plattformen (z.B. XML oder Web Services) sogar noch akzentuiert.

Die Herausforderung zur Unterstützung flexibler Datenmodelle in unterschiedlichen Systemumgebungen wird in MDA mit einer aufgeteilten Architektur beantwortet. Damit soll es MDA ermöglichen, plattform-unabhängige Modelle zu entwickeln, auf deren Basis für unterschiedliche Plattformen plattform-spezifische Modelle erstellt werden können. Basis von MDA bilden dabei die oben eingeführten und etablierten OMG-Standards UML und XMI sowie XML. Softwareentwicklung auf der Basis der MDA umfasst drei Schritte:

1. Entwurf eines domänenspezifischen Daten- bzw. Anwendungsmodells – unabhängig von irgendeiner Implementierung: Platform Independent Model (PIM).
2. Übersetzung des PIM (durch ein Werkzeug) in ein plattform-spezifisches Modell (PSM). Ein PSM ist an eine bestimmte Technologie (Betriebssystem, Programmiersprache, Middleware, Applikationsserver etc.) angepasst.
3. Übersetzung des plattform- bzw. technologiespezifischen PSM in Quellcode.

3.3. Die W3C-Standards rund um XML

Das World Wide Web Consortium (W3C) wurde im Oktober 1994 gegründet, um das Potential des WWW möglichst optimal zu erschliessen. Dazu entwickelt das W3C Protokolle, welche die Weiterentwicklung fördern und die Interoperabilität sicherstellen sollen. Das W3C hat ungefähr 400 Mitglieder aus der ganzen Welt und hat einen massgebenden Einfluss auf die Weiterentwicklung des Webs. Zurzeit sind beim W3C über 40 Normierungsprojekte in Arbeit. XML und die darauf aufbauenden Standards spielen bei der Weiterentwicklung des WWW eine zentrale Rolle, was durch Abbildung 2 sehr schön illustriert wird.

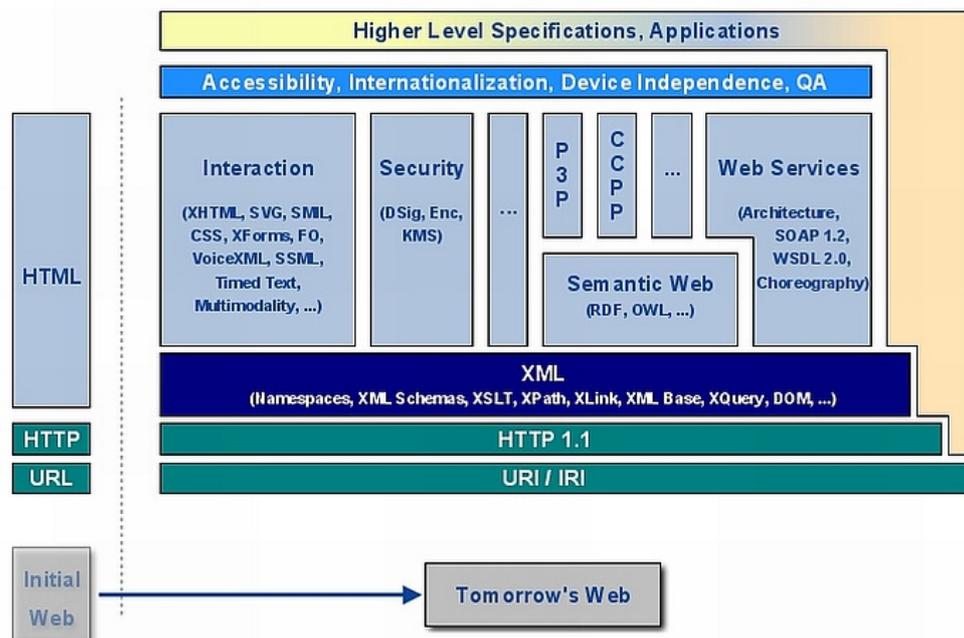


Abbildung 2: Evolution des WWW von den drei Basistechnologien URL, HTTP und HTML zu einer mächtigen Architektur auf der Basis von XML (Quelle: <http://www.w3.org/Consortium>)

In der Folge werden diejenigen W3C-Standards aus dem XML-Umfeld kurz präsentiert, welche in INTERLIS, in GML oder in beiden Standards zum Einsatz kommen.

3.3.1. XML

Die Extensible Markup Language (XML) ist in ihrem Kern ein flexibles Textformat, welches von SGML (ISO 8879) abgeleitet wurde. XML wurde ursprünglich entwickelt, um die zukünftigen Anforderungen im Bereich der elektronischen Publikation zu erfüllen. Mittlerweile hat sich aber XML zu einer Kerntechnologie in der Informatik mit einem enorm breiten Einsatzspektrum und einer ganzen Reihe von Co-Standards weiter entwickelt (vgl. auch Abbildung 2).

Aus technischer Sicht ist XML eine Sammlung von Regeln zur Schaffung von Markup- oder Auszeichnungssprachen. Damit ist XML eine Struktur beschreibende Metasprache, mit welcher eigentliche Sprachen (z.B. SVG, GML oder auch XHTML) definiert werden können. Mit XML können sowohl Inhalt als auch die Struktur von Dokumenten und allgemeinen Datensätzen beschrieben werden.

3.3.2. XML Schema

Mit XML alleine können Daten und Dokumente flexibel und bis zu einem gewissen Grad selbsterklärend codiert werden. Es wird jedoch ein zusätzlicher Mechanismus benötigt, um das dazugehörige Datenmodell formal zu definieren. Im Falle von XML werden dazu entweder die Dokumenttyp-Definition (DTD) oder neuerdings XML Schema eingesetzt. Im Gegensatz zur DTD, welches für die Modellierung von Dokumenten konzipiert wurde, ist XML Schema auch für allgemeine Daten mit komplexeren Strukturen geeignet.

XML Schema erlaubt die Beschreibung eines Datenmodells durch

- die Definition des zulässigen Vokabulars (Elemente und Attribute) und
- die Festlegung von Hierarchie, Reihenfolge und Vorgaben für die verschiedenen Elemente des Vokabulars.

Mit Hilfe eines solchen Schemas können nun Struktur und Inhalt eines entsprechenden XML-Dokuments geprüft werden. Ein grosser Vorteil ist dabei, dass diese Validierungsfunktionalität in den verbreiteten XML-Werkzeugen standardmässig enthalten ist. Da XML Schemas in XML geschrieben ist, kann sogar die Spezifikation des Datenmodells selbst auf ihre formale Korrektheit überprüft werden.

Hinweis: Im XML-Umfeld werden die Begriffe Datenmodell, Anwendungsschema und Auszeichnungssprache oft als gleichbedeutend verwendet.

3.3.3. XLink

Die XLink-Spezifikation (XML Linking Language) definiert Elemente zur Beschreibung von Links, welche das ursprüngliche Konzept der Hyperlinks als Basis des World Wide Web in die XML-Welt integrieren und gleichzeitig deutlich erweitern. XLink verwendet die XML-Syntax, um sowohl einfache Links, vergleichbar zu den einfachen Hyperlinks aus HTML, als auch komplexe Links zu beschreiben. XLink ist für Verweise auf entfernte Ressourcen

und Objekte insbesondere in der Geodatenmodellierung von grosser Bedeutung.

3.3.4. XSL und XSLT

XML verdankt seine Verbreitung unter anderem auch der Möglichkeit zur automatischen Verarbeitung und Transformation von Dokumenten und Datensätzen. Dabei spielen XSL und XSLT eine zentrale Rolle.

XSL – eXtensible Stylesheet Language – ist eine erweiterte und erweiterbare Stylesheet-Sprache, die speziell für XML entwickelt wurde und viel mächtiger und komplexer ist als die traditionelle Stylesheet-Sprache CSS (Cascading Style Sheets). XSL ist nicht nur eine Stylesheet-Sprache sondern unterstützt auch weitere Funktionen, wie zum Beispiel das Filtern und Sortieren von Elementen, die Umwandlung (Transformation) von XML-Dokumenten in ein anderes Format (u.a. HTML, PDF, SVG, PostScript), die Adressierung von Dokumentteilen und der Zugriff auf dieselben, die Formatierung von XML-Daten aufgrund der enthaltenen Datenwerte (z.B. rote Markierung von negativen Zahlen) und vieles mehr.

Auch XSL ist eine XML konforme Sprache, das heisst, XSL wird in XML formuliert und es muss keine neue Syntax erlernt werden. XSL besteht aus drei Sprachen.

- XPath – ist die Sprache für die Definition von XML-Dokumentteilen.
- XSLT (XSL Transformation) – ist die Sprache zur Transformation von XML-Dokumenten in andere Dokumenttypen (z.B. HTML) oder in andere XML-Dokumente.
- XSL-FO (XSL Formatting Objects) – ist eine Sprache zur Definition der Darstellung von XML. Sie kommt zum Einsatz bei der Umwandlung der Resultate einer XSL Transformation und ermöglicht beispielsweise die direkte Generierung von PDF-Dokumenten aus XML-Daten.

XSLT und XPath wurden als zwei separate W3C-Empfehlungen publiziert. Die Definition der XSL Formatting Objects ist ein Bestandteil der XSL Empfehlung.

3.3.5. Schematron

Schematron passt nur bedingt in diese Zusammenstellung, da dieser Mechanismus nicht von der W3C entwickelt wird. Schematron wurde am Academia Sinica Computing Centre in Taipei, Taiwan entwickelt (Academia Sinica Computing Centre, 2003) und durchläuft zurzeit die Standardisierung zur zukünftigen ISO-Norm: ISO/IEC 19757 - DSDL Document Schema Definition Language - Part 3: Rule-based validation – Schematron. Schematron ist (optionaler) Bestandteil von GML3 und wird aus diesem Grund hier eingeführt.

Wie im Titel der neuen ISO-Norm bereits impliziert, soll Schematron eine **Regel basierte Validierung von XML-Dokumenten** ermöglichen. Schematron ist somit eine Schema-sprache, mit einem anderen Funktionsprinzip als DTDs oder XML Schema. DTDs und XML Schema sind vorwiegend Grammatik basiert, während Schematron eben Regel basiert ist. Ein Schematron Schema definiert eine Menge von Regeln oder Aussagen (engl. asserti-

ons), deren Einhaltung bei der Validierung eines Dokuments überprüft wird. Es ist zwar prinzipiell möglich, auch eine Grammatik in dieser Regel-basierten Art zu definieren, in der Praxis wird man jedoch ein Schematron Schema mit einer Grammatik-basierten Schema-sprache kombinieren, um so jeweils das optimale Tool für eine gegebene Fragestellung (Grammatik-Definition oder allgemeine Regeln) verwenden zu können. Schematron setzt auf XSLT auf und verwendet XPath-Ausdrücke, um die Regeln zu formulieren und benutzerdefinierte Berichte zu generieren.

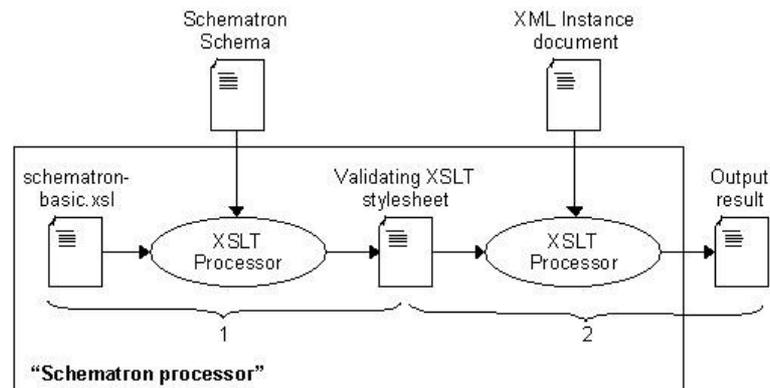


Abbildung 3 Überprüfung von XML-Instanzdokumenten mittels Schematron
(Quelle: http://www.topologi.com/resources/schtrn_xsd_paper.html)

Die Abbildung 3 zeigt eine Übersicht über den Prozess einer Überprüfung von XML Instanzdokumenten gegenüber einem Schematron Schema. Dabei ist zu beachten, dass als Werkzeuge nur ein Basis-XSL-Stylesheet (schematron-basic.xsl) und ein XSLT Prozessor (Bsp. Saxon, MSXML 4.0) benötigt werden. Den Prozess der Überprüfung bilden nachher zwei auf einander folgende XSL-Transformationen. Dieser Prozess kann problemlos "von Hand" ausgeführt werden, es existieren aber auch frei erhältliche Tools, welche diese 'Validierung' in einem Schritt ausführen. Ein solches Tool ist zum Beispiel unter <http://www.topologi.com/products/validator/index.html> erhältlich.

Einfaches Beispiel – Erweiterte Prüfung eines INTERLIS2-Transfers

Das folgende Beispiel soll aufzeigen, wie mit Schematron eine Überprüfung der Co-Existenz von zwei XML-Elementen in einem XML-Instanzdokument durchgeführt werden kann.

Auszug aus dem XML-Dokument:

```

<BspTextPos>
  <Pos>
    <COORD>
      <C1>699025.077</C1>
      <C2>262468.475</C2>
    </COORD>
  </Pos>
  <HAlI>Right</HAlI>
  <VAlI>Base</VAlI>
</BspTextPos>
  
```

Ein Schematron-Schema soll nun sicherstellen, dass die XML-Elemente 'HALi' und 'VALi' nur gemeinsam vorkommen dürfen. Das zugehörige Schematron-Schema sieht wie folgt aus:

```
<schema xmlns="http://www.ascc.net/xml/schematron"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <pattern name="Test Co-Existenz">
    <rule context="/BspTextPos">
      <assert test="(HALi and VALi) or
(not (HALi) and not (VALi))">
        HALi oder VALi fehlt
      </assert>
    </rule>
  </pattern>
</schema>
```

Das 'pattern'-Tag definiert einen Namen für diesen Test. Dieser Name (Test Co-Existenz) wird im Output aufgeführt. Mit dem 'rule'-Tag wird ein Kontextknoten mittels XPath adressiert. Dies bedeutet, dass dieser Test auf alle Knoten mit der entsprechenden Adressierung angewendet wird. Mit dem 'assert'-Tag wird schliesslich die Testbedingung eingeführt. Diese Testbedingung wird mittels XSLT-Funktionen formuliert. Der Elementinhalt von 'assert' wird nur in den Output ausgegeben, wenn der Test nicht korrekt war (false). Im obigen Beispiel wird getestet ob entweder die Elemente 'HALi' und 'VALi' gemeinsam existieren oder keines von beiden vorhanden ist. Die Syntax dieser Testregeln entspricht der XSL-Syntax. Bei der Validierung gemäss Abbildung 3 des oben aufgeführten Instanzdokumentes wird somit keine Meldung in den Output geschrieben.

Das folgende Instanzdokument produziert demnach eine Fehlermeldung:

```
<BspTextPos>
  <Pos>
    <COORD>
      <C1>699025.077</C1>
      <C2>262468.475</C2>
    </COORD>
  </Pos>
  <HALi>Right</HALi>
</BspTextPos>
```

Da jetzt das 'VALi' Element fehlt, wird folgender Output-Text ausgegeben:

Schematron Report

Test Co-Existenz

```
HALi oder VALi fehlt
  /BspTextPos[1]
  <BspTextPos>...</>
```

Dieses kleine Beispiel zeigt nur einen geringen Teil der Mächtigkeit von Schematron. Zur Definition von Testbedingungen können praktisch alle XSL-Konstrukte verwendet werden. Es kann folglich nicht nur auf Strukturelemente (Knoten, Elemente, Attribute) sondern auch auf deren Inhalt zugegriffen werden.

Bezug zu XML Schema bzw. GML

Schematron kann in XML Schema Dokumente eingebettet werden. Das heisst, dass in XML Schema nicht mögliche Konsistenzbedingungen durch Definition von Schematron-Konstrukten eingebracht werden können. Der Validierungsprozess (siehe Abbildung 4) kann anschliessend mit frei erhältlichen XML-Standard-Tools durchgeführt werden.

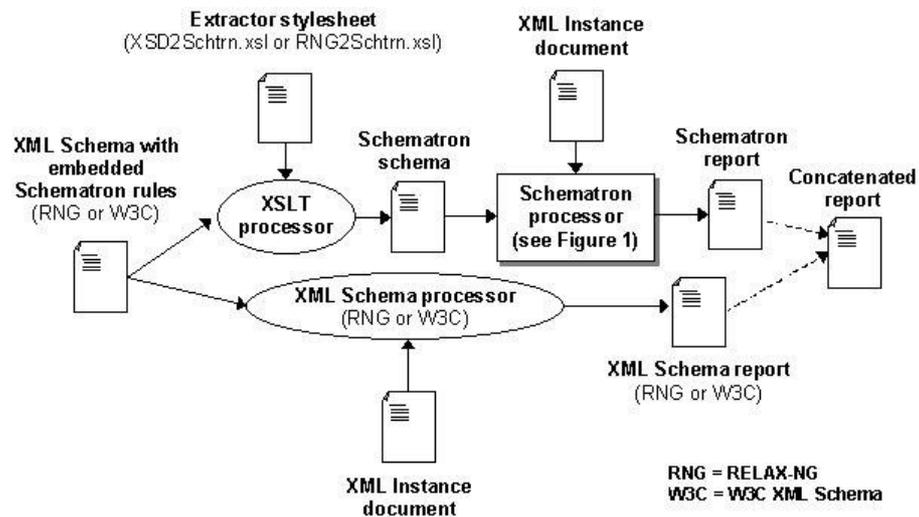


Abbildung 4 Validierung eingebetteter Schematron-Anweisungen

Die Abbildung 4 zeigt den Validierungsprozess zur Überprüfung von XML-Instanzdokumenten gegenüber einem XML Schema mit eingebetteten Schematron-Anweisungen. Dieser Prozess kann zum Beispiel auch vom oben erwähnten Schematron Validator der Firma Topologi P/L "in einem Guss" durchgeführt werden.

4. Vergleich von INTERLIS 2 und GML 3

In diesem Kapitel werden die wichtigsten Aspekte der beiden Standards aufgeführt und miteinander verglichen. Die Reihenfolge der Aufstellung hält sich in den Grundzügen an den Aufbau des Referenzhandbuchs von INTERLIS 2. In Ergänzung zur nachfolgenden Gegenüberstellung wurden einzelne Aspekte praktisch untersucht. Die Ergebnisse dieser Untersuchungen werden im darauf folgenden Kapitel präsentiert.

4.1. Standardisierung – Gremien, Prozesse und Produkte

INTERLIS 2	GML 3
Art der Norm	
<p>INTERLIS ist eine nationale 'de jure' Norm der Schweizerischen Normenvereinigung (SNV):</p> <p>INTERLIS 1: SN 612030 (Vermessung und Geoinformation - INTERLIS Modellierungssprache und Datentransfermethode)</p> <p>INTERLIS 2: SN 612031 (Vermessung und Geoinformation, INTERLIS 2 Modellierungssprache und Datentransfermethode)</p>	<p>GML ist zurzeit eine Spezifikation eines Industriekonsortiums (OGC) und bildet somit eine 'de facto'-Norm.</p> <p>Es sind Harmonisierungsbestrebungen mit der International Organization for Standardization (ISO) im Gang. Es ist insbesondere vorgesehen, GML als ISO Norm 19136 (Geographic Information – Geography Markup Language GML) zu etablieren. Als Zieldatum für die offizielle Publikation als ISO-Standard wird der 15.5.2005 angegeben (Stand Januar 2004).</p>
Wie ist der Standard spezifiziert und dokumentiert?	
<p>INTERLIS 2 ist im Referenzhandbuch (KOGIS, 2003) beschrieben, welches einen Umfang von ca. 180 Seiten aufweist.</p> <p>Seit kurzem existiert in Ergänzung dazu eine sehr gut geschriebene Einführung in die Modellierung raumbezogener Daten mittels UML und INTERLIS (Dorfschmid und Brawer, 2003).</p>	<p>Die Spezifikation von GML 3.0 (Open GIS Consortium, 2003a) umfasst ca. 530 Seiten. Nach einer teils recht ausführlichen Beschreibung von GML auf ca. 240 Seiten, folgt der normative Teil (Anhänge A, B, C), welcher Regeln zur Bestimmung der Konformität und die GML Basisschemas beinhaltet. Dieser normative Teil der Spezifikation umfasst etwas mehr als 110 Seiten. Die restlichen ca. 170 Seiten (Anhänge D bis G) beinhalten Beispiele zu verschiedenen Anwendungen von GML, welche als informativ bezeichnet werden.</p> <p>Die Basisschemas von GML stehen auch online zur Verfügung.</p> <p>Zu beachten sind die Abhängigkeiten von GML zu anderen Standards, wie zum Beispiel XML-Schema und XLink.</p>

INTERLIS 2	GML 3
Wer bildet das Normierungsgremium?	
<p>Das Normierungsgremium ist die Schweizerische Normenvereinigung (SNV). Zuständig für INTERLIS 2 ist das INB / TK 151.</p> <p>Der Standard wurde mehrheitlich vom "INTERLIS 2-Kernteam" (siehe Vorwort INTERLIS 2 Referenzhandbuch) entwickelt. Seit Anfang 2002 liegt die Verantwortung für die Redaktion des Referenzhandbuches bei der KOGIS (Koordinationsstelle für Geoinformation und geographische Informationssysteme des Bundes).</p>	<p>Das Normierungsgremium ist zurzeit das Open GIS Consortium (OGC), ein Verbund von ungefähr 260 Firmen, Hochschulen und Regierungsstellen.</p> <p>Der Standard wurde mehrheitlich von den Autoren Simon Cox, Paul Daisey, Ron Lake, Clemens Portele und Arliss Whiteside entwickelt. Diese bilden im Wesentlichen das "GML-Kernteam".</p> <p>Auf Seite der ISO ist das TC 211 zuständig für die Harmonisierung mit der ISO 19100-Serie.</p>
Kontext / Umfeld (andere Standards)	
<p>INTERLIS ist Bestandteil einer Familie von SNV-Normen zum Thema Vermessung und Geoinformation:</p> <p>SN 612010 Vermessung-Informatiksicherheit – Sicherheit und Schutz von Geodaten</p> <p>SN 612020 Datenreferenzmodell GEOBAU</p> <p>SN 612030 INTERLIS Modellierungssprache und Datentransfermethode</p> <p>SN 612031 INTERLIS 2 Modellierungssprache und Datentransfermethode</p> <p>SN 612040 Gebäudeadressen - Struktur, Georeferenzierung, Darstellung und Datentransfer</p>	<p>GML ist Bestandteil einer ganzen Serie von Geonormen des OGC. Dabei bauen beispielsweise die folgenden Spezifikationen auf GML auf:</p> <ul style="list-style-type: none"> • Web Feature Service (WFS) und • OpenGIS Location Services (OpenLS). <p>Eine detaillierte Aufstellung aller OGC-Standards ist unter www.opengis.org zu finden.</p>
Fazit	
<p>INTERLIS ist ein etablierter 'de jure'-Standard mit einem eher bedächtigen, praxisgerechten Entwicklungsrhythmus und mit einer vorwiegend nationalen Verbreitung. GML demgegenüber wird durch Mitglieder eines internationalen, mehrheitlich US-dominierten Industriekonsortiums – mit teilweise stark divergierenden Interessen – sehr schnell weiter entwickelt.</p>	
<p>Mit dem angelaufenen Einbezug von GML in den Normierungsprozess der ISO besteht die Chance, das aktuelle, (zu) rasche Entwicklungstempo etwas zu drosseln, um damit neben der internationalen Verbreitung auch eine tatsächliche Umsetzung in der Praxis zu ermöglichen. Bei einer Etablierung von GML als ISO-Norm dürfte aber der langfristige Bestand konkurrenzierender nationaler Standards in Frage gestellt sein.</p>	

4.2. Grundprinzipien

4.2.1. Modellierungskonzept

INTERLIS 2	GML 3
Was ist das grundlegende Modellierungskonzept?	
<p>INTERLIS verfolgt den Ansatz der objekt-orientierten, modell-basierten und plattform-unabhängigen Geodatenmodellierung. INTERLIS 2 wurde gegenüber INTERLIS 1 vor allem mit neuen objekt-orientierten Konzepten ergänzt. Ausserdem basiert neu ein möglicher Austauschmechanismus von INTERLIS auf XML.</p>	<p>GML verfolgt den Ansatz der objekt-orientierten, modell-basierten und plattform-unabhängigen Geodatenmodellierung. Allerdings ist GML abhängig von den W3C-Standards rund um XML (speziell XML-Schema).</p>
Welche Modellierungssprache(n) / Datenbeschreibungssprache(n) kommen zum Einsatz?	
<p>In INTERLIS können Datenmodelle in insgesamt 3 Repräsentationen abgebildet werden:</p> <ul style="list-style-type: none"> • INTERLIS-Beschreibung (normativ) • UML-Klassendiagramm (informativ) • XML-Applikationsschema (informativ) <p>Die INTERLIS- und UML-Repräsentationen werden in INTERLIS-2 als weitgehend gleich berechtigt behandelt und können weitgehend verlustfrei (UML/INTERLIS-Editor) hin und her transformiert werden.</p> <p>Das XML-Applikationsschema demgegenüber wird in INTERLIS als abgeleitetes Produkt und nicht als vollwertige Modellbeschreibung betrachtet.</p>	<p>In GML finden die folgenden 2 Modelldarstellungen Verwendung:</p> <ul style="list-style-type: none"> • GML-Applikationsschema (normativ) • UML-Klassendiagramm (informativ) <p>Die Modellbeschreibung erfolgt in GML primär in der Form von XML Schema. Allerdings sind sämtliche GML-Basis-Schemas, wie zum Beispiel feature.xsd oder topology.xsd auch in der Form von UML-Klassendiagrammen dokumentiert.</p> <p>Im April 2004 wurde die GML-Version 3.1 veröffentlicht. In dieser Version sind neu normative Abbildungsregeln zwischen UML-Klassendiagrammen (nach ISO 19109) und GML-Applikationsschemen spezifiziert (Anhang E und F). Allerdings können GML-Applikationsschemen nach wie vor von Hand erstellt werden, bzw. mit anderen Hilfsmitteln generiert werden.</p>

4.2.2. Objekt-Konzept

INTERLIS 2	GML 3
Objekte und Klassen	
<p>Ein Objekt in INTERLIS besteht aus den Daten eines Gegenstandes der realen Welt und ist eindeutig identifizierbar. Typischer-</p>	<p>Abstrahierte Phänomene der Realwelt werden in GML als Feature bezeichnet. Ein Feature kann als Typ (Abstraktion) oder</p>

INTERLIS 2

weise besitzen zahlreiche Objekte gleichartige Eigenschaften und können daher zusammengefasst werden. Eine Menge von Objekten (Objektmenge) mit gleichartigen Eigenschaften wird als **Klasse** bezeichnet. Jeder Eigenschaft entspricht (mindestens) ein Attribut.

GML 3

auch als Instanz (Repräsentation) vorkommen.

Eine Feature Collection, also eine Sammlung von Features, kann selber auch als Feature betrachtet werden.

Features können geometrische, aber auch andere Eigenschaften (properties) haben, welche durch Attribute definiert werden.

Eigenschaften von Objekten bzw. Klassen

Mit der Beschreibung einer Klasse in INTERLIS wird unter anderem festgehalten, welche Eigenschaften oder Merkmale die einzelnen Objekte tragen. Diese werden **Attribute** genannt.

Die Attributwerte der einzelnen Objekte sind dabei nicht beliebig, sondern müssen bestimmten **Bedingungen** genügen, die zur Beschreibung eines Attributes gehören.

Der Zustand eines Features ist durch ein Set von Eigenschaften (Name, Typ und Wert) definiert. Typ-Definitionen definieren die Anzahl von Eigenschaften (zusammen mit dem Namen und Werten), die ein Feature haben darf.

Ein GML-Feature kann eine beliebige Anzahl von Eigenschaften aufweisen. Die Attributwerte können gewissen Restriktionen unterliegen.

Datentypen

INTERLIS bietet für die Beschreibung von Objekteigenschaften eine Reihe von **Basisdatentypen** an (Zeichenketten, numerische Datentypen, Aufzählungen, kartesische und ellipsoidische 2D- bzw. 3D-Koordinaten, Linien, Flächen), auf deren Basis neue, **komplexe Datentypen** definiert werden können.

Ein Attribut kann sowohl einen Basis-Datentypen als auch komplexe Unterstrukturen enthalten, die nur im Zusammenhang mit ihrem Hauptobjekt existieren und auch nur über dieses auffindbar sind.

Als GML Datentypen können die Standard-Typen von XML-Schema (ca. 42 Stück), die in den GML-Basis-Schemen vordefinierten Datentypen und eigene eingeschränkte oder erweiterte Typen verwendet werden. Eine Typendefinition kann dabei im ganzen Namensraum verwendet werden.

Konsistenzbedingungen

Attributwerte müssen in INTERLIS den Anforderungen des jeweiligen Typs entsprechen. Zusätzlich können weitere Konsistenzbedingungen formuliert werden.

Attributwerte in GML müssen den Typ-Definitionen entsprechen. Zudem können weitere Einschränkungen (restrictions) definiert werden.

Konsistenzbedingungen werden in Abschnitt 4.3.5 ausführlicher behandelt.

Klassenhierarchien

INTERLIS 2 unterscheidet zwischen den drei Stufen Modell (Schlüsselwort MODEL), Thema (TOPIC) und Klasse (CLASS).

Da eine Sammlung von Features (engl. FeatureCollection) wiederum als einzelnes Feature angesehen werden kann, ist die

INTERLIS 2

Damit weisen Datenmodelle in INTERLIS eine fixe 'Tiefe' auf, was die softwaremässige Umsetzung erleichtern dürfte.

GML 3

Hierarchie von GML-Datenmodellen im Grundsatz 'beliebig tief' und vom jeweiligen Datenmodell abhängig. In der Praxis dürfte es sinnvoll sein, die Anzahl Hierarchiestufen möglichst gering zu halten oder diese sogar zu beschränken. Eine beliebige 'Tiefe' der Hierarchiestufen kann zu schwer lesbaren bzw. zu schwer verständlichen Modellen führen.

Beziehungen

Beziehungen zwischen Objekten werden in INTERLIS 2 als eigenständige Konstrukte, so genannte Beziehungsklassen bzw. Assoziationsklassen beschrieben. Die Eigenschaften dieser Beziehungen sind deren Kardinalität, deren Stärke (Assoziation, Aggregation oder Komposition) und deren Rolle.

Beziehungen zwischen GML-Features werden entweder durch das direkte Einsetzen des untergeordneten Objektes in das übergeordnete Element (Stichwort 'by value') oder mit der XML-Hilfssprache XLink ('by reference') beschrieben.

Beziehungen werden im Abschnitt 4.3.2 im Detail behandelt.

Fazit

INTERLIS und GML weisen ein ähnliches objekt-orientiertes und plattform-unabhängiges Modellierungskonzept auf. Beide Standards benutzen UML-Diagramme als Ergänzung zu den jeweiligen normativen Modellbeschreibungen, wenn auch zurzeit noch mit unterschiedlicher Verbindlichkeit. Die Struktur des INTERLIS-Datenmodells ist mit seiner vorgegebenen drei-stufigen Hierarchie etwas starrer als GML mit seiner beliebigen Hierarchietiefe, dürfte aber gerade deswegen in der Praxis einfacher umzusetzen sein.

4.3. Ausgewählte Aspekte beider Standards

4.3.1. Beschreibungssprache

INTERLIS 2

Syntax der Beschreibungssprache

In INTERLIS wird die Beschreibungssprache im Standard formal definiert. Die INTERLIS-Beschreibungssprache basiert dabei auf einer ergänzten EBNF (erweiterte Backus-Naur-Notation).

Diese textuelle Repräsentation ist durch Menschen gut lesbar, kann aber auch durch speziell dafür geschriebene Parser gut interpretiert werden.

GML 3

In GML wird als normative Beschreibungssprache XML Schema in Kombination mit anderen Auszeichnungssprachen (z. Bsp. XLink, Schematron) verwendet, was eine zusätzliche formale Definition der Beschreibungssprache erübrigt.

XML Schema ist – wie XML selbst – zwar grundsätzlich durch Menschen lesbar, aber primär für eine möglichst einfache Verar-

INTERLIS 2

Da Software zur Interpretation dieser speziellen Notation ('INTERLIS-Parser') extra programmiert werden muss, ist die Anzahl der verfügbaren Tools relativ bescheiden. Immerhin bieten die Parser-Klassen des frei verfügbaren INTERLIS-Compilers eine gewisse Erleichterung bei der Umsetzung eigener Softwarelösungen.

GML 3

beitung durch Computer entwickelt.

Das Lesen und Interpretieren von XML Schema wird durch eine Reihe von XML-Parsern in Programmiersprachen oder in vollständigen Softwaretools unterstützt. Dabei kann auf eine grosse Zahl an frei erhältlichen Tools aus den unterschiedlichsten Anwendungsbereichen zurückgegriffen werden.

Trennung von Datenmodell, Codierungsregeln und Daten

In INTERLIS besteht eine strikte Trennung zwischen dem Datenmodell (Datenbeschreibung in INTERLIS), den Codierungsregeln (z. Bsp. in XML Schema) und den Transferdaten (z. Bsp. in XML).

Da das Datenmodell weitgehend in XML Schema mit abgebildet wird, führt dies zu einer gewissen Redundanz und zu potentiellen Diskrepanzen zwischen der Modellbeschreibung und den Codierungsanweisungen für den Datentransfer. (In den Untersuchungen trat denn auch genau dieses Problem auf.)

Die Unabhängigkeit von Modellbeschreibung und Codierungsregeln hat den Vorteil, dass der Transfermechanismus ohne Anpassungen der Modellierungssprache und der existierenden Modelle vollständig ausgewechselt werden kann.

In GML sind das Datenmodell (in XML Schema) und die Daten (in XML) ebenfalls getrennt. Durch die Verwendung von XML Schema enthält das Datenmodell auch gleich die (zumeist impliziten) Codierungsregeln für die darauf basierenden XML-Transferdateien.

Die Verwendung von XML Schema als Datenbeschreibungssprache hat den Vorteil der Redundanzfreiheit zwischen Modellbeschreibung und Codierungsregeln. Sie bringt aber mit sich, dass GML von XML und damit von dessen zukünftiger Entwicklung abhängig wird. Transformationsmechanismen wie XSL dürften aber einen weitgehend automatisierten Übergang auf zukünftige XML-Versionen – oder sogar auf allfällige Nachfolgestandards – ermöglichen.

Ein positiver Aspekt dürfte die Tatsache sein, dass für die aktuelle Lösung mit XML 'nur' ein Mechanismus erlernt und eingesetzt werden muss – wenn auch ein sehr umfassender und komplexer.

Wie ist das Basis-Datenmodell definiert?

Das Basis-Datenmodell von INTERLIS wird über eine Sammlung von unterstützten **Sprachelementen** und einen Satz von Regeln definiert.

Das vollständige Basismodell von GML 3 ist definiert durch **28 XML-Schemas**, welche nach Bedarf importiert und verwendet werden können.

4.3.2. Assoziationen / Beziehungen

INTERLIS 2	GML 3
Objektbeziehungen	
<p>INTERLIS unterscheidet zwei Arten von Beziehungen zwischen Objekten: eigentliche Beziehungen und Referenzattribute.</p> <p>Eigentliche Beziehung: Menge von Objektpaaren, werden als eigenständige Konstrukte, so genannte Beziehungsklassen (bzw. Assoziationsklassen), beschrieben, die auch selbst wieder erweiterbar sind.</p> <p>Mit einem Referenzattribut wird der Bezug von einem Objekt bzw. einem Strukturelement zu einem anderen Objekt geschaffen. Eine solche Beziehung ist nur dem referenzierenden, nicht aber dem referenzierten Objekt bekannt. Sie ist also einseitig.</p> <p>INTERLIS 2 unterstützt nicht nur Zweierbeziehungen, sondern erlaubt auch mehrfache Beziehungen und solche mit eigenen Attributen. Eine Beziehungsklasse ist damit auch selbst wieder eine Objektklasse.</p>	<p>GML unterscheidet zwischen zwei Arten von Beziehungen: "by value" und "by reference".</p> <p>By value: Der Wert eines Objektes wird direkt als "property" eingesetzt. Der Name dieses Elementes entspricht dem Rollennamen der Beziehung. Vordefinierte GML-Muster (z. Bsp. gml:member) können verwendet werden. Diese Art von Beziehungsmodellierung entspricht einer Komposition (untergeordnetes Objekt ist voll abhängig von seinem "Ganzen").</p> <p>By reference: Der Wert eines Objektes wird durch einen Link repräsentiert. GML benützt hier die Auszeichnungssprache XLink zur Codierung solcher Beziehungen. Dies entspricht in seiner einfachsten Form einer gerichteten Assoziation. (Mit XLink wären auch symmetrische binäre Beziehungen oder Mehrfachbeziehungen möglich.) Dieser Mechanismus ist vergleichbar mit dem Konzept der Referenzattribute in INTERLIS.</p>
Unterstützte Beziehungstypen	
<p>INTERLIS 2 unterscheidet bei der Modellierung zwischen Assoziation, Aggregation und Komposition. In allen Fällen sind die beteiligten Objekte individuell ansprechbar. Bei Assoziation und Aggregation existieren die beteiligten Objekte unabhängig voneinander. Bei Aggregation und Komposition gibt es eine Asymmetrie zwischen den beteiligten Klassen. Allerdings ist die Codierung dieser verschiedenen Typen in den Datensätzen nicht vollständig gelöst.</p>	<p>In GML (bzw. XML-Schema) ist die Komposition ("by value") die grundlegende Beziehungsart. Nicht existenzabhängige Beziehungen (Aggregation, Assoziation) werden durch geeignete Konstrukte ersetzt (Referenzattribut per XLink).</p>

4.3.3. Geometrie und Topologie

INTERLIS 2	GML 3
Geometriemodell	
<p>Das Geometriemodell von INTERLIS unterscheidet zwischen Punkten (COORD,</p>	<p>Das Geometriemodell von GML folgt demjenigen vom Topic 1 der OGC Abstract</p>

INTERLIS 2

0d), Linienzügen (POLYGON, 1d), Einzelflächen (SURFACE, 2d) und Gebietseinteilungen (AREA, 2d). Es werden keine 3d-Oberflächendarstellungen oder 3d-Volumenobjekte unterstützt.

Alle geometrischen Attributtypen basieren auf der Repräsentation von Koordinatenwerten, welche in INTERLIS durch den Attributtyp COORD codiert wird.

GML 3

Specification (entspricht ISO DIS 19107, Geographic Information – Spatial schema).

Unter anderem können Punkte (Point, 0d), Linienzüge (_Curve, 1d), Flächen (_Surface, 2d) und Volumenkörper (_Solid, 3d) modelliert werden.

Grundlage für alle diese Basisgeometrietypen, ist die Repräsentation von Koordinatenwerten, welche in GML 3.0 in zwei verschiedenen Arten codiert werden können: gml:pos und gml:coordinates. (Details vgl. GML-Spezifikation)

Unterstützung von Gebietseinteilungen

INTERLIS bietet dazu den Attributtyp AREA an. Dabei besteht eine Gebietseinteilung (AREA) aus einer endlichen Menge von (allgemeinen) Flächen und Restflächen, welche die Ebene überlappungsfrei überdecken.

Zur Sicherstellung einer AREA sind in der INTERLIS-Spezifikation folgende Konsistenzbedingungen definiert:

- Linienzüge einer Gebietseinteilung müssen immer echte Grenzlinien sein. Es dürfen also keine Linienzüge existieren, bei denen auf beiden Seiten die gleiche Fläche liegt. Dies ist auch durch die Definition der Fläche ausgeschlossen.
- Liegen auf beiden Seiten eines Linienzuges definierte Gebietsobjekte, muss jedes Kurvenstück (Verbindung zweier Stützpunkte) des einen Gebietsobjektes in Geometrie und Attributen genau einem Kurvenstück des anderen Gebietobjektes entsprechen.

GML bietet verschiedene Ansätze zur Modellierung von Gebietseinteilungen an. Ein möglicher Ansatz ist die Modellierung mit dem Typ 'gml:Surface'. Eine 'Surface' ist immer ein homogener, endlicher Teil einer Ebene. Dieser Teil einer Ebene wird durch 'gml:Polygon'-Elemente in einzelne Flächen (engl. patches) unterteilt. Somit können einzelne Gebietsobjekte auf die 'gml:Polygon'-Elemente referenzieren.

Im Anhang 2 – Beispiel AREA in GML ist eine mögliche Variante zur Modellierung des Datentyps AREA in GML aufgeführt. Allerdings ist diese Konstruktion komplex und aufwändig zu realisieren (siehe auch 5.2.2).

Overlaps

In INTERLIS ist das Konzept von Overlaps integriert. Dabei kann eine definierte Toleranz für Überschneidungen von verschiedenen Linienelementen (Bsp. Kreisbogen) des **gleichen** Linienzuges gewährt werden.

Das Konzept von Overlaps und Toleranzen wird in GML nicht behandelt.

INTERLIS 2	GML 3
3D-Geometrie	
INTERLIS beinhaltet zwar 3-dimensionale Koordinatenwerte, aber keinen Attributtyp für 3D-Oberflächendarstellungen oder 3D-Volumenkörper	GML bietet in den Basisschemen 'geometryPrimitives.xsd' und 'geometryAggregates.xsd' Anwendungsmuster für 3D-Objekte an. Diese bauen auf den Flächentypen von GML auf.
Topologie	
INTERLIS 2 unterstützt kein explizites Topologiemodell.	Der Topologie-Unterstützung in GML liegt das Thema 1 der OGC Abstract Specification (entspricht ISO DIS 19107) zugrunde. Topologische und geometrische Beziehungen können mit GML in bis zu 3 Dimensionen modelliert werden. Dazu bietet GML die topologischen Elemente Knoten (engl. Nodes), Kanten (Edges), Flächen (Faces) und Volumenkörper (Solids).
Coverages	
Wird in INTERLIS 2 nicht unterstützt.	Ein Coverage ist ein spezieller Featuretyp, welcher eine räumliche Ausdehnung einem homogenen Wertebereich zuordnen kann. Ein Coverage wird zur Modellierung und zur Darstellung von räumlichen Beziehungen und der geometrischen Ausdehnung von geografischen Phänomenen benutzt. Mögliche Anwendungsformate sind Rasterförmige Daten, Dreiecksvermaschungen (TIN), punktbezogene Daten (Grids) und polygonförmige Daten. Insbesondere in den Bereichen Fernerkundung, Meteorologie, Tiefen-, Höhen-, Boden- und Vegetationskartografie werden Coverages häufig verwendet.

4.3.4. Einheiten und Bezugssysteme

INTERLIS 2	GML 3
Einheiten / Units	
Basiert auf SI-Normen (SI = Système international d'unités) bestehend aus Basiseinheiten (base units), und abgeleiteten Einheiten (derived units)	Basiert auf SI-Normen, bestehend aus Basiseinheiten (base units), und abgeleiteten Einheiten (derived units)
Koordinatensysteme	
INTERLIS 2 unterstützt die Verwendung unterschiedlicher Koordinatensysteme. Der Verweis auf das entsprechende Koordina-	GML benötigt den Verweis auf ein Koordinatensystem (CRS) immer, wenn Koordinaten-Informationen verwendet werden.

INTERLIS 2

tensystem kann in INTERLIS im DOMAIN-Abschnitt bei der Definition der Koordinatentypen angebracht werden (Beispiel siehe Testdatensatz).

Hinweis: Die Unterstützung von Koordinatensystemen ist zurzeit **kein normativer** Bestandteil der INTERLIS 2-Spezifikation (siehe informativen Anhang J, INTERLIS 2-Referenzhandbuch).

GML 3

Die Referenzierung wird mit Hilfe des Attributs 'srsName' gemacht. Dieses steht für alle Geometrie Elemente in GML zur Verfügung.

Hinweis: Auch in GML ist die Codierung von Koordinatensystemen noch **nicht normativ** geregelt (siehe informativen Anhang D, GML 3.0-Spezifikation). Es existieren aber Basisschemen, welche als Entwurf zu betrachten sind, voraussichtlich in der nächsten Version (3.1) werden die CRS-Schemen normativer Bestandteil von GML sein.

Es existiert ein "Recommendation Paper" des Open GIS Konsortiums zur Kodierung von CRS-Definitionen. Das 'srsName'-Attribut kann auf bereits bestehende (z. Bsp. EPSG) oder auf eigene Bibliotheken verweisen. Diese Sammlungen von CRS-Definitionen müssen jedoch der oben erwähnten Empfehlung des OGC entsprechen.

4.3.5. Konsistenzbedingungen

INTERLIS 2

Wie kann die Datenkonsistenz sichergestellt werden?

Attributwerte müssen in INTERLIS den Anforderungen des jeweiligen Typs entsprechen. Zusätzlich können weitere Konsistenzbedingungen formuliert werden. Folgende Typen werden dabei im INTERLIS 2-Referenzhandbuch erwähnt:

- MandatoryConstraint
- PlausibilityConstraint
- ExistenceConstraint
- UniquenessConstraint

GML 3

Attributwerte in XML Schema müssen den Typ-Definitionen entsprechen; weitere Einschränkungen (restrictions) können definiert werden. Um weiterführende Konsistenzbedingungen zu definieren, kann die Auszeichnungssprache Schematron verwendet werden.

Konsistenzbedingungen, die sich auf ein einzelnes Objekt beziehen

INTERLIS unterscheidet hier Bedingungen, die jedes Objekt einer Klasse erfüllen muss ("**harte**" Konsistenzbedingungen, 'MandatoryConstraint'), und Bedingungen, deren Verletzung zwar an sich möglich, aber selten ist ("**weiche**" Konsistenzbedingungen, 'PlausibilityConstraint').

Konsistenzbedingungen lassen sich in GML entweder mit der Standard-Modellierungssprache XML Schema oder mit Hilfe der Zusatzsprache Schematron einführen.

Mit Schematron können unter anderem "harte" Konsistenzbedingungen, welche

INTERLIS 2

GML 3

sich auf ein einzelnes Objekt beziehen eingeführt werden.

Die Erfassung von "weichen" Konsistenzbedingungen ist nach jetzigem Erkenntnisstand in GML nicht möglich.

Plausibilitätsbedingung

Mittels einer Plausibilitätsbedingung kann angegeben werden, welcher Prozentanteil der Instanzen der Klasse die Bedingung normalerweise erfüllen soll (vgl. Regel PlausibilityConstraint).

Ist zurzeit in GML 3 nicht vorgesehen.

Existenzbedingung

Mit der Existenzbedingung (Regel ExistenceConstraint) kann gefordert werden, dass der Wert eines Attributes jedes Objektes der Bedingungs-Klasse in einem bestimmten Attribut einer Instanz anderer Klassen vorkommt.

Überprüfungen von Existenzen von Attributwerten in anderen Objekten können mittels Schematron-Anweisungen integriert werden.

Eindeutigkeitsbedingung

Eindeutigkeitsforderungen werden in INTERLIS mit dem Schlüsselwort UNIQUE eingeleitet (Regel Uniqueness-Constraint).

In XML Schema können die Schlüsselworte ID, KEY und UNIQUE für die Erstellung von Eindeutigkeitsbedingungen verwendet werden.

Fazit

INTERLIS und GML bieten eine Reihe von Konstrukten zur Beschreibung von raumbezogenen Objekten mit ihren Eigenschaften und gegenseitigen Beziehungen. Die Modellierung zweidimensionaler Objekte ist von wenigen Ausnahmen abgesehen in etwa gleichwertig. INTERLIS hebt sich mit der Modellierung von Overlaps und der Spezifikation von Plausibilitätsbedingungen von GML ab. Andererseits bietet GML Funktionalität, welche in INTERLIS zurzeit nicht vorhanden ist, wie zum Beispiel die Modellierung von 3D-Objekten oder die Unterstützung unterschiedlicher Coverage-Typen.

4.4. Transfer

4.4.1. Domänenspezifische Datenmodelle bzw. Applikationsschemas

INTERLIS 2	GML 3
Welche Rolle spielen domänenspezifische Datenmodelle?	
<p>Domänenspezifische Datenmodelle ermöglichen eine Verständigung zwischen den beteiligten Fachleuten über Umfang und Art der Beschreibung ausgewählter raumbezogener Phänomene.</p> <p>Domänenspezifische Datenmodelle sind die Quintessenz des modellbasierten Datenaustauschs mit INTERLIS.</p> <p>Heute existieren über 100 domänenspezifische Modelldefinitionen in INTERLIS. Diese sind aber noch überwiegend in INTERLIS 1 formuliert. Einige dieser Datenmodelle, wie Beispielweise das 'Datenmodell 01 der amtlichen Vermessung', sind rechtlich verbindlich.</p>	<p>Domänenspezifische Datenmodelle werden in GML als Applikationsschemas bezeichnet.</p> <p>Die Erkenntnis über die enorme Bedeutung dieser Applikationsschemas für den erfolgreichen Praxiseinsatz von GML scheint erst langsam zu reifen. Applikationsschemas und Applikationssprachen (vgl. unten) sind denn auch ein Hauptthema der laufenden GML-Aktivitäten bei OGC (Stand: Okt. 2003).</p> <p>Zurzeit existieren noch wenig publizierte Applikationsschemas für GML 3.0. Ein Beispiel eines solchen Schemas ist das 'Mineral Occurrence Application Schema des Geological Survey of Canada' (Galdos Systems Inc., 2002)</p>
Wie können Datenmodelle bzw. Applikationsschemas erzeugt werden?	
<p>Die Erstellung von Datenmodellen erfolgt entweder manuell mit der Beschreibungssprache INTERLIS (.ili) oder durch UML-Klassendiagramme (Bsp. UML/INTERLIS-Editor).</p>	<p>Die Spezifikation erlaubt die Erstellung von Applikationsschemas mittels manueller Programmierung in XML Schema, durch Erstellung von UML-Klassendiagrammen, oder durch andere Tools.</p>
Unterstützung hierarchisch aufgebauter Datenmodelle (z. Bsp. Basismodell Bund und darauf aufbauende kantonale Modelle mit Mehranforderungen)	
<p>Mittels der objekt-orientierten Konzepte der (Einfach-) Vererbung und des Polymorphismus können Datenmodelle erweitert und wieder verwendet werden.</p> <p>Datensätze welche kantonale Mehranforderungen beinhalten, sind gemäss Referenzhandbuch auch gegenüber den übergeordneten Datenmodellen (Bsp. Bundesmodell) gültig. Eine Überprüfung auf der Basis von XML ist hier allerdings nur mit speziellen INTERLIS-Prüfwerkzeugen mög-</p>	<p>GML (bzw. XML-Schema) unterstützt ebenfalls das Konzept der (Einfach-) Vererbung. Die GML-Basisschemas stellen an sich schon ein Basismodell dar, von welchem ein Applikationsschema abgeleitet werden kann. Von diesem Schema kann wiederum ein Applikationsschema abgeleitet werden, welches detailliertere Angaben zu Objekten enthält. Die Tiefe dieser Hierarchie ist (wie in INTERLIS) beliebig erweiterbar.</p>

INTERLIS 2	GML 3
lich.	
Wie erfolgt die Prüfung von Datenmodellen?	
<p>Die Prüfung von INTERLIS-Datenmodellen erfolgt mit dem INTERLIS-Compiler, welcher kostenlos zur Verfügung steht.</p> <p>Der Compiler verfügt unter anderem über folgende Funktionen:</p> <ul style="list-style-type: none"> • syntaktische formale Prüfung des INTERLIS-Datenmodells • Generierung eines XML-Datenschemas 	<p>Die Prüfung von GML-Datenmodellen erfolgt durch die Validierung des Applikationsschemas mit den GML-Basissschemen. Diese Überprüfung kann mit einem beliebigen, validierenden XML-Tool durchgeführt werden. Mit demselben Tool können auch XML-Instanzdokumente (Daten) im Rahmen der Möglichkeiten von XML Schema überprüft werden.</p> <p>Um Schematron-Anweisungen (vorwiegend Konsistenzbedingungen) zu überprüfen kann ein beliebiger XSL-Prozessor eingesetzt werden.</p>

Fazit

GML basiert mit XML Schema auf einem allgemeinen IT-Standard, der durch ein grosses Spektrum an Softwaretools unterstützt wird. Datenmodelle in XML Schema können durch diese Softwaretools sehr einfach geprüft werden, sind aber für Menschen ohne diese Hilfsmittel nur schwierig zu interpretieren. Demgegenüber sind INTERLIS-Modelle auch durch Menschen gut lesbar und nachvollziehbar – ohne eine Umwandlung in UML zwingend erforderlich zu machen. Die maschinelle Prüfung von INTERLIS-Datenmodellen und -Datensätzen erfordert aber eine Software mit einem speziell entwickelten 'INTERLIS-Parser'.

4.5. Neuerungen in GML 3.1

Die vorliegende Studie bezieht sich auf die veröffentlichte und offiziell gültige Version 3.0 von GML. Beim OGC ist aber die Definition von GML 3.1 in vollem Gang und liegt als "Recommended Paper" vor. Ein wichtiger Aspekt der laufenden Arbeiten ist die Harmonisierung von GML mit der ISO-Normenserie 19100. Das Ziel dieser Harmonisierungsbestrebungen ist die Erstellung der ISO-Norm 19136 (Geography Markup Language GML). Gemäss der aktuellen Veröffentlichung auf www.iso.org befindet sich diese Norm zurzeit in der Vorbereitungsphase (Committee Stage). Das Zieldatum einer definitiven Anerkennung wird mit Mitte 2005 (15-05-2005) angegeben.

Obwohl GML 3.1. erst als Entwurf vorliegt, soll hier ein kurzer Überblick auf sich abzeichnende Änderungen bzw. Neuerungen gegeben werden:

Eine wichtige Neuerung ist die Spezifikation von gegenseitigen Abbildungsregeln zwischen UML- und GML-Applikationsschemen. Dabei soll die Konformität zu den ISO-Normen 19109 (Rules for application schema) und 19118 (Encoding) hergestellt werden. Dies soll

die Benutzung von UML-Klassendiagrammen zur Modellierung in GML ermöglichen; allerdings bleiben auch "handgeschriebene" oder mit anderen Hilfsmitteln erstellte GML-Applikationsschemen als normative Modellrepräsentationen bestehen. Das heisst, dass die normative Modellbeschreibung weiterhin in XML Schema verfasst ist (analog INTERLIS Modellbeschreibung in INTERLIS-Sprache), aber eben zum Beispiel UML-Klassendiagramme zur Modellierung eingesetzt werden können. Ob diese Abbildung streng nach ISO-Regeln in GML spezifiziert wird, kann erst anhand der offiziellen Veröffentlichung von GML 3.1 bzw. der ISO-Norm 19136 beurteilt werden.

Im Rahmen der Harmonisierung werden auch diverse GML-Typen umbenannt, oder sogar neu eingeführt. Allerdings haben die Arbeiten auch zum Ziel, GML nicht noch mehr zu erweitern, und somit die Komplexität nicht weiter zu erhöhen. Ob dies gelingt, kann auch erst bei der offiziellen Veröffentlichung beurteilt werden.

Die Harmonisierungsbestrebungen zwischen OGC und ISO sind sicher begrüßenswert. Wie schnell diese Harmonisierung gelingt und wie deren definitives Resultat aussieht, ist zurzeit aber noch schwer abzuschätzen.

5. Praktische Untersuchungen

In diesem Kapitel wird eine Reihe von praktischen Untersuchungen präsentiert. Diese stammen zu einem Grossteil aus der Diplomarbeit von Adrian Annen, welche er im Herbst 2003 an der Abteilung Vermessung und Geoinformation der FHBB durchgeführt hat.

Im Rahmen dieser Untersuchungen wurden folgende Fragestellungen analysiert:

- Modellierung eines ausgewählten INTERLIS-Themas in GML 3 (Ebene Einzelobjekte aus dem Datenmodell AV01) als Basis für die nachfolgenden Untersuchungen
- Bestimmung von Abbildungsregeln zwischen INTERLIS 2 und GML 3 auf der Ebene Datenschema und auf der Ebene Daten
- Identifikation von allfälligen Inkompatibilitäten zwischen INTERLIS 2 und GML 3
- Untersuchung verschiedener Konvertierungsvarianten (gemäss Abbildung 5)

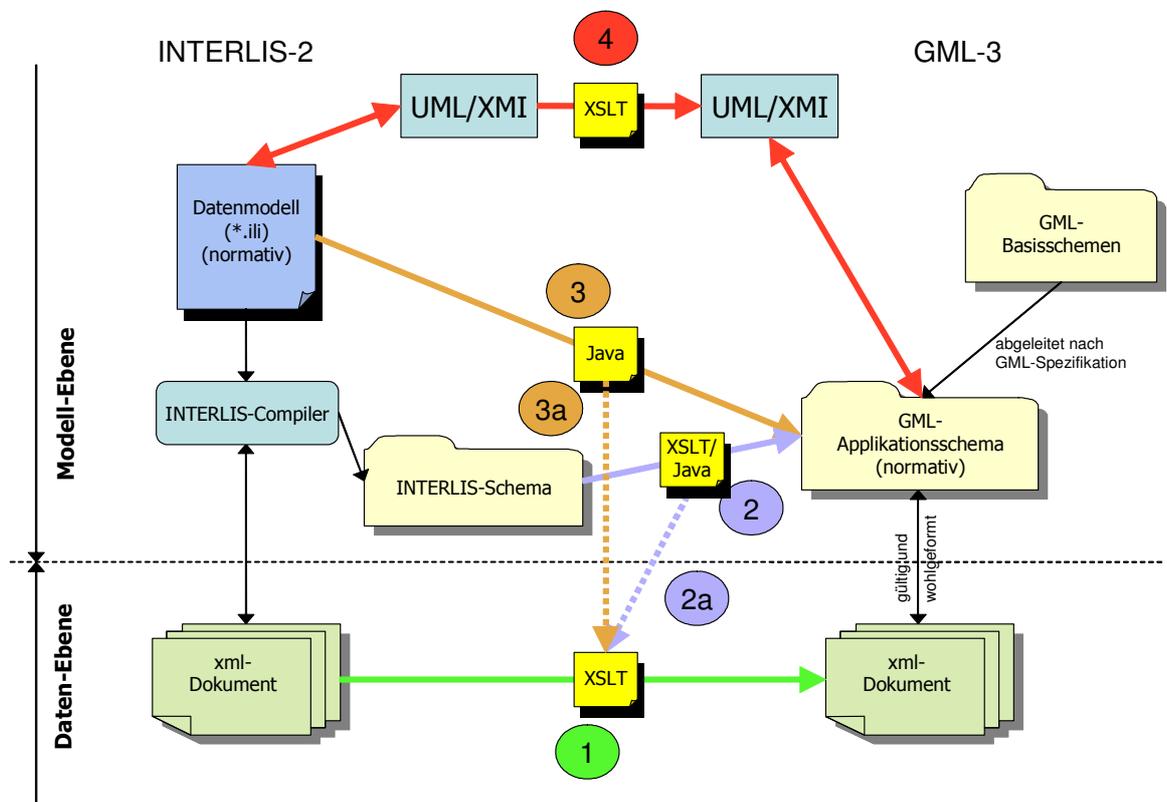


Abbildung 5: Untersuchte Konvertierungsvarianten zwischen INTERLIS 2 und GML 3

Die untersuchten Konvertierungsvarianten umfassen:

- **Variante 1** – Eine direkte Datenkonvertierung mittels XSLT (dient primär zur Identifikation der Abbildungsregeln und zur Analyse des erforderlichen XSL Codes)
- **Variante 2** – Eine modell-basierte Transformation zwischen INTERLIS XML-Schema und GML-Applikationsschema mittels XSLT und/oder Java und die direkte Ableitung des XSL Codes für die Transformation der entsprechenden Daten (2a)
- **Variante 3** – Die modell-basierte Transformation zwischen den beiden normati-

ven Modellbeschreibungen, der INTERLIS-Beschreibung und dem GML-Applikationsschema mittels Java und wiederum die Ableitung des XSL Codes für die Transformation der entsprechenden Daten (3a)

- **Variante 4** – Die Modelltransformation zwischen INTERLIS und GML mittels UML und XMI.

5.1. Modellierung eines INTERLIS-Themas in GML 3

5.1.1. Untersucher Ausschnitt aus dem INTERLIS 2-Testdatensatz

Für die nachfolgenden praktischen Untersuchungen wurde das Thema Einzelobjekte des offiziellen Testdatensatzes zur INTERLIS Version 2 ausgewählt. Das in INTERLIS 2 definierte Datenmodell wurde analysiert und der gewählte Ausschnitt mit Hilfe des UML/INTERLIS-Editors als UML-Klassendiagramm (siehe Abbildung 6) dargestellt.

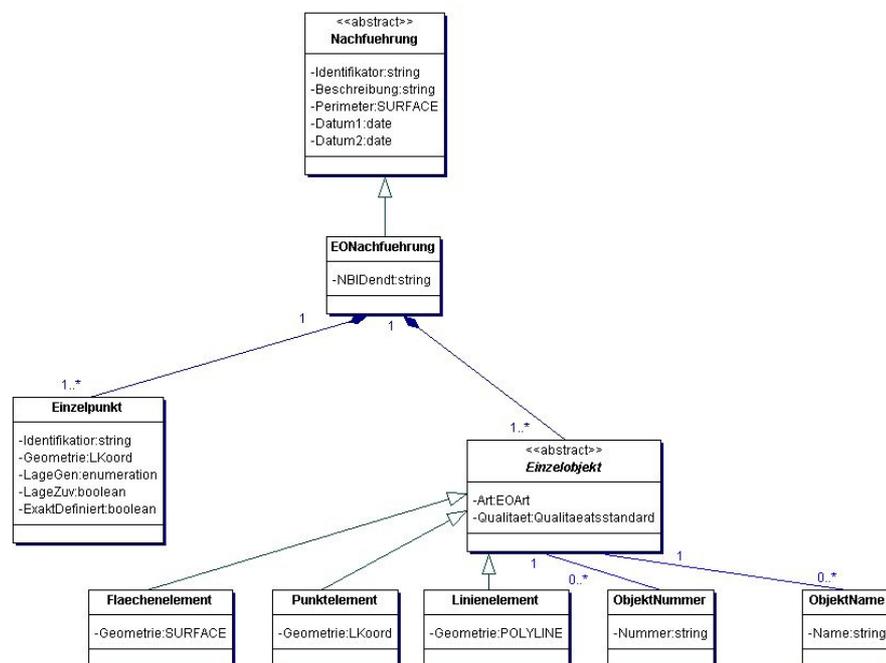


Abbildung 6: UML-Klassendiagramm des Themas Einzelobjekte

Hinweis: Der INTERLIS-Testdatensatz im XML-Transferformat enthält auch Daten für das Thema Einzelobjekte. Ein Vergleich der Struktur dieser Daten mit dem XML-Schema, welches durch den INTERLIS-Compiler erzeugt wird, ergab zum Zeitpunkt der Untersuchung (Oktober 2003) eine Reihe von Diskrepanzen zwischen Schema und Testdatensatz. Diese Unstimmigkeiten sind in Abschnitt 5.3 detaillierter aufgeführt. Inzwischen wurden diese Probleme behoben (Stand: Juni 2004).

5.1.2. Entsprechendes Datenschema in GML 3

Das Resultat der Modellierung einer bestimmten Anwendung in GML wird als Applikationsschema bezeichnet. Ein GML-Applikationsschema ist ein XML-Schema, welches die Konformitätsregeln des GML-Standards befolgt. Diese Regeln werden im folgenden Kapitel behandelt.

Im vorliegenden Fall wurde für die Thematik 'Einzelobjekte' mit Hilfe eines XML-Editors 'manuell' ein möglichst gleichbedeutendes Schema in GML 3 erstellt. Dieses GML-Applikationsschema ist in Abbildung 7 aus Gründen der Übersichtlichkeit in grafischer Form dargestellt.

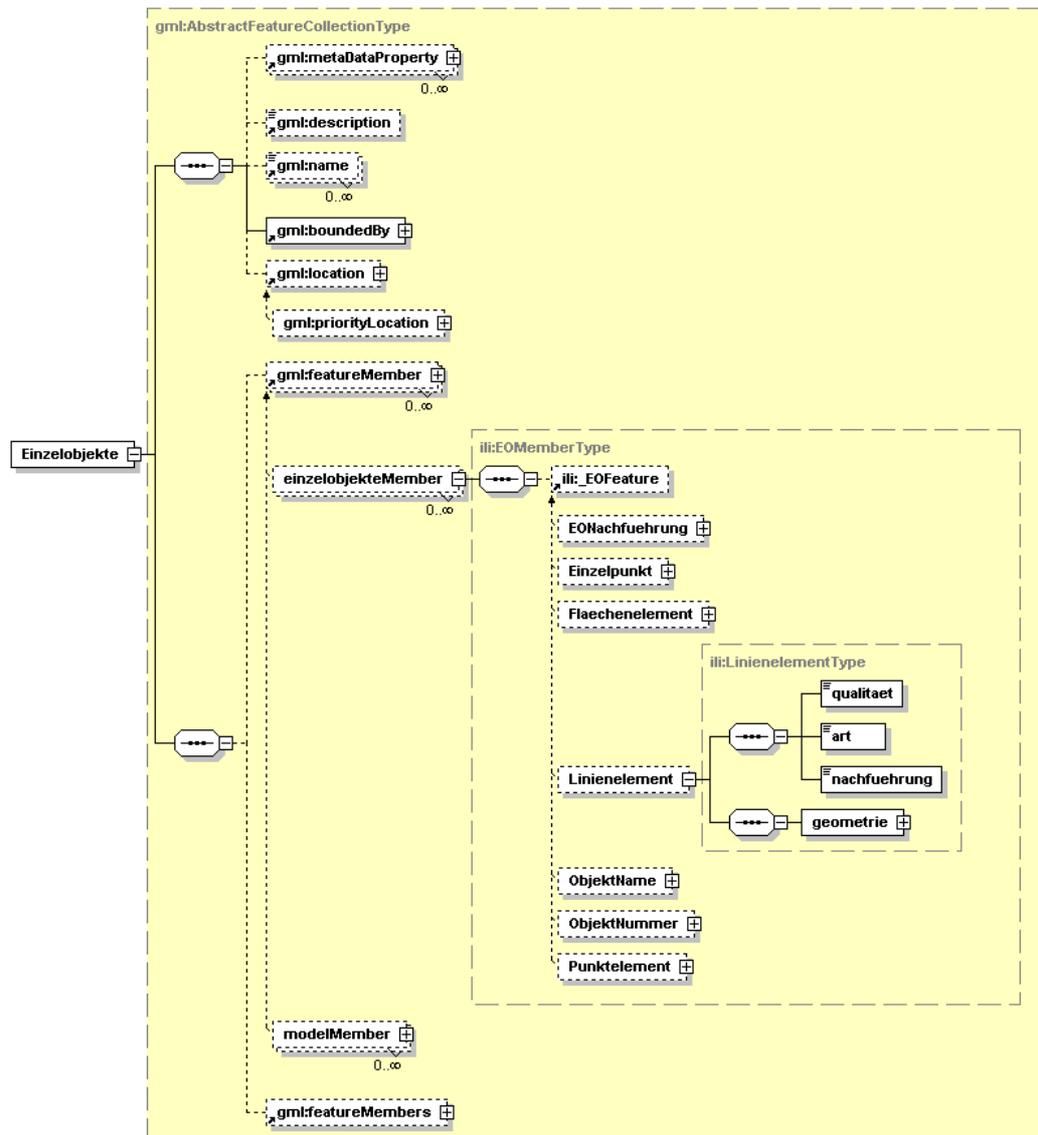


Abbildung 7: Auszug aus dem GML-Applikationsschema für das Thema Einzelobjekte des INTERLIS-Testdatensatzes (halbgrafische Darstellung im Schema Design-Modus des XML-Editors XMLSpy)

Das folgende XML-Listing enthält einen Auszug aus dem GML-Applikationsschema. Dabei wird im ersten Abschnitt der abstrakte Typ '_Einzelobjekt' definiert. Im zweiten Abschnitt wird dann die Klasse 'Linienelement' als eigentliches XML-Element definiert. Die Typdefinition basiert auf dem abstrakten Typ '_Einzelobjekt' und ererbt damit seine Kindelemente 'qualitaet', 'art' und 'nachfuehrung'. Zusätzlich wird das Kindelement 'geometrie' hinzugefügt. Diese Definition entspricht im wesentlichen der Modellierung der Klasse 'Linienelement' im Thema Einzelobjekte des Testdatensatzes zur INTERLIS Version 2. (Hinweis: Aus Übersichtsgründen wurde hier die Beziehung zur Klasse 'Nachfuehrung' als Referenzattribut modelliert, und nicht als Kompositionsbeziehung.)

```
<!-- ~~~~~~
Class: Einzelobjekt (abstract)
~~~~~ -->
<xsd:element name="_Einzelobjekt"
  type="ili:AbstractEinzelobjektType" abstract="true"
  substitutionGroup="gml:_Feature"/>
<xsd:complexType name="AbstractEinzelobjektType" abstract="true">
  <xsd:complexContent>
    <xsd:extension base="ili:AbstractEOFeatureType">
      <xsd:sequence>
        <xsd:element name="qualitaet"
          type="ili:QualitaetsstandardType"/>
        <xsd:element name="art" type="ili:EOArtType"/>
        <xsd:element name="nachfuehrung"
          type="gml:StringOrRefType"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<!-- ~~~~~~
Class: Linienelement
~~~~~ -->
<xsd:element name="Linienelement" type="ili:LinienelementType"
  substitutionGroup="ili:_EOFeature"/>
<xsd:complexType name="LinienelementType">
  <xsd:complexContent>
    <xsd:extension base="ili:AbstractEinzelobjektType">
      <xsd:sequence>
        <xsd:element name="geometrie"
          type="gml:CurveArrayPropertyType"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Hinweis: Es sollte beachtet werden, dass die Erstellung eines raumbezogenen Datenmodells in GML 3 ein Abstraktionsprozess ist, der einen gewissen Interpretations-Spielraum offen lässt und daher entsprechende Vereinbarungen erfordert. Ein bestimmter Sachverhalt kann somit – wie auch in INTERLIS – durchaus unterschiedlich modelliert werden. Durch das Aufstellen von Abbildungsregeln sollten sich aber INTERLIS-Datenmodelle einheitlich auf GML übertragen lassen.

5.2. Abbildung zwischen INTERLIS 2 und GML 3

Ziel dieser Untersuchungen war es, einerseits Abbildungsregeln zwischen INTERLIS 2 und GML 3 aufzustellen und andererseits allfällige Inkompatibilitäten oder Probleme aufzudecken. Dabei stand vorerst der Weg von INTERLIS nach GML im Vordergrund; nicht zuletzt wegen der rasch zunehmenden Umfangs von GML, welcher eine vollständige Abbildung auf INTERLIS in Frage stellt.

Zudem sollten diese Abbildungsregeln mit der Thematik 'Einzelobjekte' verifiziert werden.

5.2.1. Abbildungsregeln

Bei der Erstellung von Abbildungsregeln zwischen INTERLIS 2 und GML 3 wurde zwischen der Abbildung der Struktur und der Abbildung der Datentypen unterschieden.

Abbildung der Struktur

Die allgemeinen Regeln zur Abbildung der Modellstruktur lauten:

- Die Bedingungen für die GML-Konformität müssen immer eingehalten werden (gemäss GML 3.0-Spezifikation S. 231 ff)
- Ein INTERLIS-Modell (MODEL) entspricht einem gml:Applikationschema bzw. dem Element gml:FeatureCollection als Wurzelement eines Datensatzes
- Ein Thema (TOPIC) entspricht einer gml:FeatureCollection
- Eine Klasse (CLASS) entspricht einem gml:featureMember
- Wenn immer möglich sollen GML-Muster (engl.: Pattern) oder XML:Standardtypen verwendet werden
- Beziehungen (Assoziationen) werden in GML über Objektreferenzen realisiert (xlink:href)
- Einfachvererbungen sind zu übernehmen und modellgetreu zu definieren (Mehrfachvererbung wird weder in INTERLIS noch in GML bzw. XML-Schema unterstützt)
- Datentypen sind gemäss separater Empfehlung (siehe unten) abzubilden

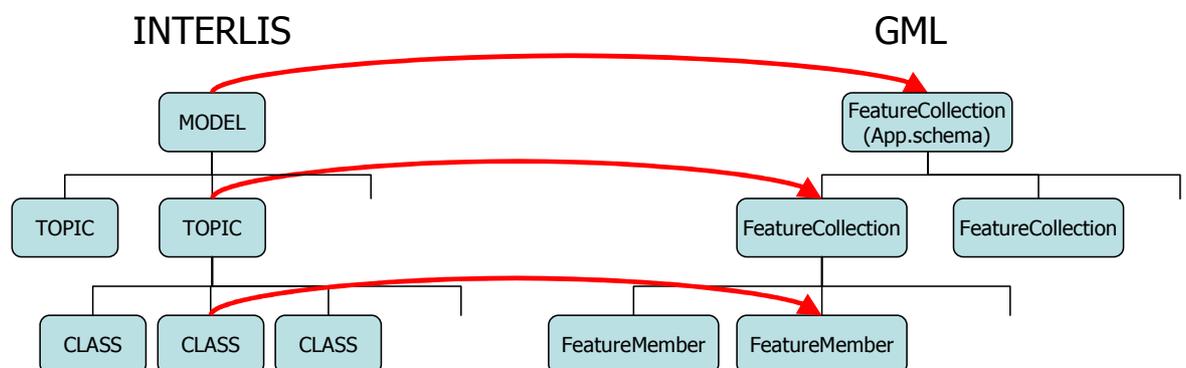


Abbildung 8: Abbildung der Modellstruktur von INTERLIS auf GML

Datentypen-Zuordnung

INTERLIS bietet eine Reihe von einfachen Basistypen (TextType, EnumerationType etc.) bis hin zu Geometriotypen (POLYLINE, SURFACE etc.) an, welche in GML eine Entsprechung haben sollten. Hierzu wurde zunächst abgeklärt, ob GML äquivalente Datentypen anbietet. Wenn kein 'eins-zu-eins' Datentyp gefunden wurde, wurde eine möglichst präzise Alternative gewählt. Insbesondere bei den Geometriotypen sind zum Teil mehrere Abbildungen möglich. Als Beispiel sei hier der Datentyp 'POLYLINE' erwähnt. Ein Linienzug kann beispielsweise in INTERLIS nur aus linearen Elementen ('STRAIGHTS') bestehen. Dies würde in GML einem 'gml:LineString' entsprechen. Da aber meistens auch Bögen ('ARCS') zugelassen werden, wurde für die Empfehlung der Datentyp 'gml:CurveArray' gewählt. Folglich handelt es sich bei der vorliegenden Zuordnung um eine **mögliche** Variante. Die entsprechende Zuordnungstabelle befindet sich im Anhang (Kap. 9).

5.2.2. Abbildungsprobleme

Im Verlauf der Untersuchungen wurden diverse potentielle Schnittstellenprobleme identifiziert. Die folgenden Ausführungen gelten wiederum für die Abbildung von INTERLIS-Modellen in GML, da die umgekehrte Abbildung beliebiger GML-Applikationsschemas nach INTERLIS den Rahmen dieser Untersuchungen gesprengt hätte.

Modellierung von Beziehungen

Über die Modellierung von Objektbeziehungen in GML bzw. XML-Schema sind in der Fachliteratur zum Teil gegensätzliche Meinungen vorhanden. Weit verbreitet ist die Meinung, dass der Mechanismus XML-Schema keine Möglichkeiten zur Modellierung von Objektbeziehungen beinhalte. XML-Schema selbst bietet tatsächlich nur sehr eingeschränkte Möglichkeiten zur Erfassung von Objektbeziehungen an. Aber GML zieht hier den Mechanismus von XLink heran. Diese Auszeichnungssprache stellt Konstrukte zur Verfügung, welche es erlauben, Beziehungen von eigenständigen Objekten zu modellieren. Somit können mit Hilfe der Kombination XML-Schema und XLink durchaus geeignete Datenmodelle entworfen werden.

Bei der Modellierung von Beziehungen ist zwischen den verschiedenen Beziehungstypen (Vererbung, Assoziation, Aggregation und Komposition) zu unterscheiden. In den folgenden Absätzen sind kurz einige Bemerkungen zu den einzelnen Beziehungstypen aufgeführt.

- **Vererbung:** Grundsätzlich unterstützen beide Mechanismen die Prinzipien der einfachen Vererbung. In XML-Schema bieten sich allerdings zwei verschiedene Möglichkeiten an. Einerseits kann eine Erweiterung (extension) erfolgen, andererseits eine Einschränkung (restriction). Somit können Vererbungsbeziehungen in beiden Mechanismen äquivalent modelliert werden.
- **Assoziation:** In UML können bidirektionale Beziehungen modelliert werden. Das heisst es können gegenseitige, gleichberechtigte Assoziationen zwischen Objekten modelliert werden. Auch in INTERLIS können solche Assoziationen modelliert werden. Allerdings stellt sich bei der Codierung von Daten die Frage, wie solche Beziehungen korrekt dargestellt werden können. Sind beide Objekte wirklich gleichberechtigt, muss

bei beiden Objekten ein Verweis auf das jeweils assoziierte vorhanden sein, oder es muss eine spezielle Assoziationsklasse eingeführt werden. Dies kann zu Problemen bei der Nachführung solcher Beziehungen führen. Da GML sehr stark auf die Codierung von Geodaten ausgerichtet ist, ist keine gleichberechtigte, gegenseitige Beziehungsart vorgesehen. Objektbeziehungen werden ausschliesslich mittels Aggregation und Komposition realisiert (siehe unten).

- **Aggregation** (Ganzes-Teile-Beziehung): Die Modellierung der Beziehungen erfolgt in GML über die Hilfssprache XLink. Das heisst die Objekte der einen Klasse (Teile) erhalten einen Verweis (engl. link) zu ihrem zugehörigen Objekt der anderen Klasse (Aggregat bzw. Ganzes). Damit können Beziehungen zwischen Objekten modelliert werden. Dieser Link wird in GML über ein Referenzattribut realisiert (by reference). Diese Technik entspricht der aus INTERLIS-1 bereits bekannten Formatierung und entspricht im Wesentlichen einer gerichteten Assoziation.
- **Komposition:** Eine Komposition kann in GML 3 über das direkte Einsetzen eines Objekts als Objekteigenschaft erzielt werden (by value). In INTERLIS 2 geschieht dies über das Schlüsselwort ASSOCIATION und der dementsprechenden nachfolgenden Syntax (-<#>). Allerdings ist im Testdatensatz und in der Spezifikation zu INTERLIS-2 kein Beispiel zu einer Codierung dieser Beziehungsart im Instanzdokument vorhanden.

Verschiedene Modellierungsaspekte

Im Verlauf der Untersuchungen wurden auch Unterschiede in der Behandlung der Geometrie sowie weiterer Aspekte gefunden. Diese Liste ist mit grosser Wahrscheinlichkeit noch nicht vollständig. Zu den aufgedeckten Unterschieden gehören:

- **Bogensegmente:** Beide Standards können Bogensegmente mit Bogenanfangs-, Bogenmittel- und Bogenendpunkt erfassen. Zusätzlich kann INTERLIS 2 Bogensegmente mit Anfangs- und Endpunkt sowie dem zugehörigen Radius erfassen. Diese Variante ist in GML 3 nicht vorgesehen.
- **Overlaps:** Eine Modellierung der tolerierbaren Überschneidungen bei Bogensegmenten ist in GML 3 nicht vorgesehen.
- **Nachkommastellen:** In GML 3 werden Koordinatenwerte mit den Basistypen 'gml:pos' oder 'gml:coordinates' codiert. Diese Typen basieren auf dem XML-Schema-Typen 'xsd:string', und können somit nicht nur Zahlenwerte sondern auch Zeichenketten beinhalten. Damit ist aber eine Beschränkung der Nachkommastellen nicht zu realisieren. Der Grund für die Wahl von 'xsd:string' für die Koordinatencodierung liegt im Format von geodätischen Koordinaten. Zum Beispiel können in GML 3 auch Koordinaten mit folgender Struktur codiert werden: 46°57'27"N, 7°25'55"E.
- **AREA:** Der INTERLIS-Typ AREA (Gebietseinteilungen) kann nicht eins zu eins in GML abgebildet werden. Eine Lösungsmöglichkeit ist die Modellierung über den GML-Typ 'gml:Surface'. Im Anhang 2 – Beispiel AREA in GML – ist ein Codebeispiel für die Modellierung eines einfachen INTERLIS-Modells mit einer Gebietseinteilung in GML aufgeführt.

5.3. Probleme im INTERLIS-Testdatensatz

Der INTERLIS-Compiler ermöglicht die Generierung eines XML-Schemas aus einem INTERLIS-Modell. Dieses XML-Schema sollte zu jedem Datensatz passen, welcher auf demselben INTERLIS-Modell beruht. Oder umgekehrt formuliert: ein INTERLIS-Datensatz ist nur gültig, wenn er mit dem entsprechenden XML-Schema erfolgreich validiert werden kann. Diese Prüfung kann mit jedem validierenden XML-Parser sehr einfach durchgeführt werden.

Die praktischen Untersuchungen (durchgeführt im Oktober 2003) mit dem INTERLIS-Datensatz (Version 2.2.) ergaben jedoch, dass der XML-Datensatz und das generierte XML-Schema nicht kompatibel waren und eine Reihe von Diskrepanzen aufwiesen. Es zeigte sich beispielsweise, dass in der verwendeten Version (2.1.8) des INTERLIS-Compilers gewisse Modelleigenschaften aus dem normativen Datenmodell (*.ili) nicht sauber in XML-Schema übertragen werden. Einfachstes Beispiel ist die Modellierung von optionalen Attributen (ohne Schlüsselwort MANDATORY), welche im XML-Schema als zwingend definiert sind (minOccurs=1). Ausserdem werden Modell übergreifende Beziehungen nicht vollständig korrekt modelliert (Probleme bei der Namensgebung der Elemente).

Eine exakte Übereinstimmung von XML-Schema und XML-Datensatz sind eine wichtige Voraussetzung für den effizienten Umgang mit XML und den Einsatz entsprechender Editier- und Prüfertools. Die aufgetauchten Diskrepanzen wurden daher umgehend der KOGIS gemeldet. Für die Untersuchungen hatten die Unstimmigkeiten zwischen Schema und Datensatz zeitaufwändige, manuelle Anpassungen zur Folge.

Bei einer erneuten Untersuchung des modifizierten INTERLIS-Compilers (Version 2.1.13) und des angepassten Testdatensatzes wurde festgestellt, dass die oben genannten Probleme mittlerweile bereinigt sind. Die Validierung des Testdatensatzes gegenüber einem aus dem Modell generierten XML-Schema konnte problemlos durchgeführt werden (Stand Juni 2004).

5.4. Konvertierungsvarianten

Auf Grund der Erkenntnisse bei der Grundlagenanalyse und der manuellen Abbildung eines Datenschemas sollen verschiedene (automatische) Abbildungs- bzw. Konvertierungsvarianten untersucht werden. Dabei soll die Untersuchung eine Bewertung und eine Abschätzung der Machbarkeit der einzelnen Varianten (vgl. Abbildung 5) beinhalten.

5.4.1. Variante 1 – Direkte Datenkonvertierung

Bei einer Abbildung zwischen INTERLIS und GML müssen nicht nur die Datenmodelle sondern auch die eigentlichen Daten konvertiert werden können. Mit der Untersuchung einer direkten formatbasierten und damit statischen Konvertierung von INTERLIS-Daten in GML sollte die Machbarkeit der Transformation auf Datenebene untersucht werden. Im Vordergrund stand dabei die Frage, inwiefern sich XSL (eXtensible Stylesheet Language) für diese Variante eignet. Gleichzeitig sollten damit wertvolle Erfahrungen für die programmgesteuerte Generierung von XSL-Stylesheets oder anderweitiger Transformationsregeln gewonnen werden, welche für eine modellbasierte Datentransformation erforderlich sind.

Als Grundlage für eine direkte Datenkonvertierung muss das Datenmodell in beiden Standards vorliegen bzw. vereinbart sein. Das in Abschnitt 5.1 entworfene GML-Applikationsschema (Thema Einzelobjekte) diente somit als Basis für den Entwurf eines XSL-Stylesheets. Die Erkenntnisse aus der Entwicklung dieses Prototyps sind hier kurz aufgeführt:

- Die Transformation einfacher Datentypen kann problemlos durchgeführt werden
- Bei komplexeren Geometrietypen (Bsp. Polylinien mit Bogensegmenten) sind jedoch aufwändige Operationen im XSL-Stylesheet anzubringen. Hier zeigt sich deutlich, dass XSL keine vollwertige Programmiersprache ist. Die grösste Schwierigkeit stellt die Realisierung von Fallunterscheidungen dar. Zum Beispiel müssen beim Datentyp POLYLINE alle möglichen Abfolgen von COORD und ARC richtig transformiert werden. Dies führt sehr rasch zu komplexen XSL-Transformationen.

Obwohl verschiedene Tools wie zum Beispiel XML Spy hilfreiche Ansichten zum Erstellen von XSL-Stylesheets anbieten, ist deren manuelle Erstellung sehr aufwändig. Da zwingend für jeden Knoten ein Abbildungsmuster (Template) erstellt werden muss, wird das XSL-Stylesheet bald unübersichtlich. Wie eingangs erwähnt, hat diese Methode den grossen Nachteil modellabhängig zu sein. Damit müsste für jedes Datenmodell ein neues XSL-Stylesheet entwickelt werden muss.

Interessant wird die Datentransformation mittels XSL-Stylesheet, wenn dieses aus der Modellbeschreibung automatisch abgeleitet werden kann.

5.4.2. Variante 2 – modell-basierte Transformation zwischen INTERLIS XML-Schema und GML-Applikationsschema

In der Variante 2 erfolgt die Transformation zwischen den beiden XML-Schemas – im vollen Bewusstsein, dass diese in INTERLIS und GML eine unterschiedliche Rolle spielen. In INTERLIS wird das XML-Schema mit dem INTERLIS-Compiler aus dem Datenmodell abgeleitet, in GML enthält das XML-Schema die normative Modellbeschreibung. Für die Transformation von einem XML-Schema auf ein anderes bietet sich auf den ersten Blick XSL (eXtensible Stylesheet Language) als Konvertierungsmechanismus an.

Zur Zielsetzung dieser Untersuchung gehört die Machbarkeitsprüfung:

- einer Modelltransformation mittels XSL oder mittels eigenständiger Applikation (Abbildung 5, Variante 2) sowie
- einer automatischen Generierung eines XSL-Stylesheets für die eigentliche Datenkonvertierung (Abbildung 5, Variante 2a)

Die Umsetzung dieser Variante mit XSL zeigte einige Probleme und Schwachstellen auf:

- Wie bereits weiter früher erwähnt, ist XSL keine vollwertige Programmiersprache. So sind gewisse Fallunterscheidungen nur sehr aufwändig und unübersichtlich realisierbar. Solche Fallunterscheidungen sind aber beispielsweise bei der Transformation von Geometrietypen unabdingbar.

- Ein weiteres Problem bei der Erstellung einer XSL-Transformation ergab sich aus der uneinheitlichen Typdefinition im vom INTERLIS-Compiler generierten XML Schema. So waren einige Typdefinitionen direkt in der Element-Deklaration vorhanden, während andere einen eigenen Typennamen (eigener ContentType) erhielten. Dies führte zu komplexen XSL-Stylesheets, welche unübersichtlich und fehleranfällig sind.

Auf Grund dieser Ergebnisse wurde die Umsetzung einer Java-Applikation für diese Konvertierungsvariante untersucht. Java stellt Klassenbibliotheken für die Interpretation der XML-Dokumente (XML-Parser) zur Verfügung und erlaubt eine flexible Bearbeitung und Transformation der Eingangsknoten.

Der Einsatz einer modernen, vollwertigen Programmiersprache hat deutliche Vorteile gegenüber einer alleine auf XSL basierenden Transformationslösung. Es ist zudem zu vermuten, dass gewisse komplexe Modellstrukturen alleine mit XSL nicht mehr automatisch abgebildet werden können. Die uneinheitliche Typendefinition im vorgelegenen INTERLIS-Eingangsdokument (XML Schema) erschwerte die Realisierung dieser Variante.

5.4.3. Variante 3 – modell-basierte Transformation zwischen den normativen Modellbeschreibungen

Bei dieser Untersuchungsvariante stand die (automatische) Abbildung zwischen den beiden normativen Modellbeschreibungen, also zwischen der INTERLIS-Modellbeschreibung und dem GML-Applikationsschema, im Vordergrund.

Wie schon in Variante 2 sollte auch hier eine Machbarkeitsprüfung für die folgenden zwei Teilaufgaben durchgeführt werden:

- die automatische Ableitung einer Modelltransformation (Abbildung 5, Variante 3) sowie
- die automatische Generierung eines XSL-Stylesheets für die eigentliche Datenkonvertierung (Abbildung 5, Variante 3a)

Da bei dieser Transformationsvariante als Input keine XML-Daten, sondern eine INTERLIS-Beschreibung vorliegen, können hier keine XML-Parsertools eingesetzt werden. Es werden vielmehr Softwarewerkzeuge zur Interpretation von INTERLIS Modellen ("INTERLIS-Parser") benötigt. Die durchgeführten Untersuchungen zeigten auf, dass sich anstelle der sehr aufwändigen Entwicklung eines eigenen INTERLIS-Parsers, die Verwendung der entsprechenden Klassen des frei verfügbaren INTERLIS-Compilers geradezu aufdrängt.

Für die Realisierung der **Variante 3** (Modelltransformation) müsste der INTERLIS-Compiler um einen GML-Output-Generator ergänzt werden. Dabei könnte auf das bestehende interne Metamodell des INTERLIS-Compilers zurückgegriffen werden. Der bestehende XML-Schema-Output dürfte zudem als wertvolle Vorlage dienen, um einen GML-Schema-Export relativ rasch realisieren zu können.

Der Vorteil eines GML-Schema-Generators als Bestandteil des INTERLIS-Compilers liegt im Vergleich zur Variante 2 darin, dass kein "Umweg" über ein INTERLIS-eigenes XML

Schema gemacht wird. Mit dieser direkten Transformation zwischen den normativen Modellbeschreibungen wird somit der Informationsverlust minimal gehalten werden.

Die **Variante 3a** (Datenkonvertierung bzw. Code-Generierung zur Datenkonvertierung) erfordert die Erweiterung um einen entsprechenden XSL-Generator. Die Entwicklung und das Austesten dieser Komponente dürften um einiges aufwändiger sein, als die Realisierung der Modelltransformation. Die XML-XML-Transformation zwischen INTERLIS und GML würde zudem formal korrekte Inputdaten erfordern, sprich widerspruchsfreie INTERLIS XML Schema und XML-Datensätze.

Mit dem XSL-Generator aus Variante 3a und den damit generierten XSL Style Sheets könnten bestehende INTERLIS-Datensätze mit beliebigen, frei verfügbaren XSL Prozessoren konvertiert werden. Denkbar wäre hier sogar eine on-the-fly Transformation, wie sie in vielen Web Publishing-Lösungen bereits an der Tagesordnung ist. (Operationelles Praxisbeispiel: Swiss Virtual Campus Projekt GITTA mit on-the-fly XML-Transformation der Lerninhalte auf Cocoon-Basis.)

5.4.4. Variante 4 – modell-basierte Transformation mittels UML

Der so genannte "model-driven-approach" ist auf der Seite von INTERLIS mit dem UML/INTERLIS-Editor der Firma Eisenhut bereits erfolgreich implementiert. Dieser Editor erlaubt die Modellierung von Datenschemen in UML-Klassendiagrammen (Unified Modeling Language) (vgl. 3.2.1). Auf der anderen Seite sind intensive Bestrebungen im Gang die Abbildung von UML nach GML-Applikationsschemen zu spezifizieren (siehe auch 4.5). Somit stellt sich die Frage nach der Transformation von UML-Klassendiagrammen. Da aber UML nur graphische Konventionen zur Darstellung von Datenmodellen beinhaltet, fehlt hier eine formale Beschreibungssprache, welche Klassendiagramme beschreiben kann. Im Vordergrund steht dabei die Benutzung von XMI (Xml Metadata Interchange), welches unter anderem die formale Repräsentation von UML ermöglicht (vgl. 3.2.3).

Die Realisierung einer modell-basierten Transformation via UML für den erfolgreichen und zuverlässigen Praxiseinsatz erfordert nach unserer Ansicht die **normative Festlegung der Abbildungsregeln** zwischen den aktuellen normativen Modellbeschreibungen: im Falle von INTERLIS also zwischen der INTERLIS-Modellbeschreibung und der UML-Repräsentation und im Falle von GML zwischen dem Applikationsschema und der entsprechenden UML-Repräsentation.

Diese Transformationsvariante wäre klar die eleganteste und zukunftssträchtigste Lösung. Sie setzt aber, wie schon aufgezeigt, verbindliche UML-Modellierungs- und Abbildungsvorschriften für beide Mechanismen voraus. Diese Voraussetzungen sind bei INTERLIS ansatzweise erfüllt; bei GML sind die Diskussionen über die Abbildungsregeln zwischen UML und XML Schema eben erst angelaufen (vgl. 4.5 zu den Aktivitäten im Bereich GML 3.1).

Für die zukünftigen Anwender stellt sich zudem die Frage nach der Komplexität und Transparenz der Variante 4. Neben den beiden aktuellen normativen Modellbeschreibungen INTERLIS-Beschreibung und GML-Applikationsschema kommen bei dieser Variante ein bzw. zwei weitere Mechanismen hinzu, die insbesondere im Falle von XMI doch recht komplex und abstrakt sind.

Bemerkung: Die Autoren vermuten, dass dieses Nebeneinander verschiedener, anspruchsvoller Mechanismen ein gewichtiger Grund war, weshalb sich das OGC im Falle von GML in einer Anfangsphase pragmatisch auf einen Mechanismus, nämlich XML und XML Schema beschränkt hat. Eine entsprechende Anfrage beim OGC blieb allerdings unbeantwortet.

5.4.5. Zusammenstellung und Vergleich der Varianten

In der folgenden Tabelle werden die hauptsächlichen Stärken und Schwächen der untersuchten Varianten zusammengestellt:

Variante	Vorteile / Stärken	Nachteile / Schwächen
1	<ul style="list-style-type: none"> + realisierbar + Basis für die Datenkonvertierung (und damit für alle anderen Varianten) + Konvertierungsregeln können automatisch aus der Modelltransformation abgeleitet werden (Varianten 2a, 3a, 4a) 	Bei manueller Erzeugung: <ul style="list-style-type: none"> - nicht modell-basiert - grosser Aufwand zum vorgängigen Abbilden von Datenmodellen in zwei Datenbeschreibungssprachen (INTERLIS und GML)
2	<ul style="list-style-type: none"> + modell-basiert + basierend auf Standard-XML-Technologien + automatische Generierung von XSL-Stylesheet für Datenkonvertierung grösstenteils möglich (2a) 	<ul style="list-style-type: none"> - Umweg über XML-Schema aus ILI-Compiler (Informationsverluste) - Entwicklungsaufwand / XSL 'keine' Programmiersprache - Standard-XML-Technologien sind nicht für Geodaten entwickelt worden
3	<ul style="list-style-type: none"> + modell-basiert + erlaubt Transfer zwischen normativen Modellbeschreibungen + bestehender 'INTERLIS-Parser' vorhanden (Compiler) + automatische Generierung von XSL-Stylesheets für Datenkonvertierung möglich (3a) 	<ul style="list-style-type: none"> - für den INTERLIS-Compiler muss eine neue Output-Funktion entwickelt werden - Vollständigkeit der Abbildung von INTERLIS nach GML kann erst bei einer allfälligen Implementierung abschliessend beurteilt werden. Beispiel: Codierung von 'AREA'
4	<ul style="list-style-type: none"> + voraussichtlich zukunftsweisend (MDA) + basierend auf internationalen Standards (ISO, OMG) + könnte in Zukunft Variante 3 ersetzen (sobald Schwachstellen behoben, vgl. rechts) 	<ul style="list-style-type: none"> - fehlende Abbildungsregeln zwischen normativen Modellbeschreibungen und UML (INTERLIS und GML) - neue Technologie: Fragezeichen betreffend Tools, Know-how, Praxistauglichkeit

Tabelle 1: Gegenüberstellung der Varianten zur Modelltransformation

6. Zusammenfassung und Beurteilung

In diesem Kapitel werden die Stärken und Schwächen von INTERLIS und GML, sowie deren Gemeinsamkeiten und Unterschiede zusammengestellt. Diese Zusammenstellung wird abgeschlossen durch eine Auflistung von Kernpunkten, die durchaus kontrovers beurteilt werden dürften.

6.1. INTERLIS 2 – Stärken und Schwächen

Stärken

- **Präzis und schlank** – INTERLIS ist ein schlanker und dennoch mächtiger Mechanismus. Er erlaubt eine präzise und detaillierte Beschreibung von Datenmodellen und ermöglicht damit die automatische (modell-basierte) Ableitung des Transferformats.
- **Praxiserprobte Vorgängerversion** – Die Version 1 von INTERLIS ist praxiserprobt, bewährt und gesetzlich verankert. So existieren beispielsweise über 100 Datenmodelle für die unterschiedlichsten Anwendungsgebiete.
- **Evolution statt Revolution** – Beim Wechsel auf INTERLIS 2 wurden die Elemente von INTERLIS 1 fast vollständig übernommen und erweitert. Dadurch können Datensätze relativ einfach von INTERLIS 1 nach INTERLIS 2 migriert werden.
- **Softwaretools** – Es existiert eine Reihe von Softwaretools für die operationelle Arbeit mit INTERLIS. Eine besondere Rolle kommt dabei dem INTERLIS-Compiler zu, der aus einem Datenschema in IDDL das entsprechende Transferformat ableitet und die syntaktische Richtigkeit des Datenschemas überprüft.
- **UML und Text** – Neben der traditionellen textuellen Beschreibung von Datenmodellen unterstützt INTERLIS 2 neu auch die grafische Beschreibung mittels UML-Klassendiagrammen. Auch dafür steht mit dem UML/INTERLIS-Editor ebenfalls ein kostenloses Tool zur Verfügung.

Schwächen

- **Eigene Sprache** – Die eigene Modellierungssprache INTERLIS Data Definition Language (IDDL) ist zwar durch Menschen gut lesbar, deren Unterstützung in einer Software ist aber deutlich aufwändiger als beispielsweise eine XML-basierte Repräsentation. Allerdings mildern frei verfügbare Tools wie der INTERLIS-Compiler diese Problematik zu einem gewissen Mass.
- **Nur Modellierung und Transfer** – INTERLIS ist primär auf die Normierungsaspekte Datenmodellierung und Datencodierung limitiert. Eine Realisierung von Diensten auf der Basis von INTERLIS wäre zwar technisch ohne weiteres möglich, dürfte aber auf Grund der geographisch beschränkten Verbreitung von INTERLIS in der Praxis einen sehr schweren Stand haben.

- **'Insellösung'** – INTERLIS ist zwar weitgehend kompatibel mit internationalen Standards, aber es fehlt eine wirkliche Einbettung in eine internationale Geonormenfamilie.
- **(Zu) gute Version 1** – Da INTERLIS 1 in der Praxis gut eingeführt ist und einen Grossteil der Bedürfnisse abdeckt, erfolgt die Umstellung auf INTERLIS 2 nur zögerlich. Möglicherweise scheuen die Systemhersteller und Anwender die Investitionen für eine Umstellung auf INTERLIS 2, vielleicht auch mit Blick auf die internationalen Entwicklungen im GML-Umfeld.
- **Rolle von XML (Schema)** – Das Potential des XML-basierten Transfers wird noch kaum ausgenützt. Im Laufe der Untersuchungen tauchten Diskrepanzen bei der Erzeugung von XML Schema Dokumenten durch den INTERLIS-Compiler auf. Dies lässt darauf schliessen, dass INTERLIS 2 und insbesondere dessen XML-Transfer in der Praxis noch wenig eingesetzt werden oder dass zumindest das Potential von XML noch nicht genutzt wird. Mittlerweile sind die in dieser Untersuchung aufgedeckten Probleme behoben worden und es kann davon ausgegangen werden, dass in Zukunft die Rolle von XML Schema im INTERLIS-Umfeld aufgewertet wird.
- **Wie weiter mit Erweiterungen?** – INTERLIS steht vor der Herausforderung, weitere nicht-Standard-Datentypen und räumliche Konzepte zu unterstützen. Dazu gehören Aspekte wie 3D-Geometrien (Oberflächenrepräsentationen und Körper), Topologie, Raster und unregelmässig verteilte Punktwolken. Dabei stellt sich die Frage, ob und wie weit sich ein paralleler Effort zu OGC und ISO rechtfertigen lässt.

6.2. GML 3 – Stärken und Schwächen

Stärken

- **Marktmacht** – Wohl die grösste Stärke von GML dürfte die Marktmacht der involvierten Normierungsgremien OGC und mittlerweile auch ISO, sowie der weltweit angelaufenen Aktivitäten sein.
- **Pragmatische Modellierungssprache** – Die Wahl von XML Schema als normative Datenmodellierungssprache ist zwar wenig elegant und ist gewissen Einschränkungen unterworfen; aber sie dürfte die softwaremässige Umsetzung begünstigen, da nur ein Mechanismus unterstützt werden muss.
- **Normenfamilie** – GML ist in eine Familie von Geonormen eingebettet, sowohl bei OGC als auch zukünftig bei ISO. So existieren bereits erste Spezifikationen für Dienste auf GML-Basis (z.B. OGC Web Feature Service).
- **Applikationssprachen** – Mit GML wird das Konzept von Applikationssprachen unterstützt, welche auf GML aufbauen und dessen Sprachelemente erweitern und / oder einschränken können.

Schwächen

- **Versionen-Flut** – Eine Hauptschwäche von GML ist sicherlich das (zu) rasche

Entwicklungstempo mit 3 Vollversionen in nur 3 Jahren! Hoffnung auf eine gewisse Konsolidierung machen einzig die Harmonisierungsbestrebungen mit ISO.

- **Umfang und Komplexität** – GML 3 besteht aus ca. 28 Basis-Schemas, die je nach Art und Komplexität des zu modellierenden Sachverhaltes importiert und somit verwendet werden können. Dabei besteht die Gefahr, dass die meisten Implementierungen nur Basis-Profile unterstützen werden. Hier besteht somit ein Bedarf nach klaren Konventionen innerhalb eines Anwendungsbereichs (z.B. Amtliche Vermessung Schweiz) über ein verbindlich zu unterstützenden Profils.
- **Einschränkungen von XML Schema (?)** – Ein oft geäussertes Mangel besteht in der Wahl von XML Schema als Modellierungsmechanismus. GML behebt einige der Hauptschwächen von XML Schema durch die Integration von zusätzlichen XML-Mechanismen wie XLink für die Modellierung von Beziehungen und Schematron für die Regel basierte Validierung von Datensätzen. Dadurch werden aber weitere, externe Mechanismen hinzugezogen, was die Komplexität des Standards wiederum erhöht.
- **Kryptische Modellbeschreibungen** – Die Verwendung von XML Schema führt zwar zu formal strengen Modellbeschreibungen, die mit Hilfe von Softwaretools einfach gelesen und geprüft werden können. Für Menschen sind aber GML-Applikationsschemas nur sehr schwer und mit viel Übung lesbar.
- **Fehlende Praxis** – Zurzeit existieren für GML, insbesondere für GML 3 noch wenig domänenspezifische Datenmodelle (GML-Applikationsschemas) und praktische Datensätze. Existierende Datensätze in GML 2 basieren zumeist noch nicht auf einer modellbasierten Philosophie – sie stammen oft aus dem System X, dürfen in ihrer Struktur nicht verändert werden und können nur so wiederum in das System X importiert werden (vgl. auch Herstellersupport).
- **(Noch) mangelnder Herstellersupport** – Die praktische Unterstützung von GML durch die namhaften Hersteller von GI-Software ist heute noch sehr rudimentär und hinkt den entsprechenden Werbebotschaften sowie dem Normierungsprozess stark hinterher. So gibt und gab es kaum eine nennenswerte Unterstützung für GML 1, nur eine rudimentäre, erste Unterstützung von GML 2 mit eingeschränkter bis fehlender modell-basierter Abbildung und noch kaum eine Unterstützung für GML 3.

6.3. Wichtigste Gemeinsamkeiten

INTERLIS und GML weisen trotz vieler Unterschiede eine Reihe von Gemeinsamkeiten auf. Die wichtigsten davon sind:

- Eine Basisphilosophie, welche die modell-basierte Geodatenmodellierung grundsätzlich ermöglicht.
- Ein objekt-orientiertes Modellierungskonzept mit den Grundkonstrukten Packages, Klassen, Attributen und Beziehungen sowie einer Einfachvererbung.
- Die Unterstützung domänenspezifischer Applikationsschemata in XML Schema.
- Eine Codierung des Datentransfers auf der Basis von XML, welche ein einfaches Hand-

ling und eine weit reichende formale Prüfung mit Standard-XML-Tools ermöglicht.

6.4. Wichtigste Unterschiede

Zu den wichtigsten Unterschieden zwischen INTERLIS und GML gehören:

- Zwei unterschiedliche normative Modellierungssprachen: INTERLIS basierend auf der gleichnamigen Modellierungssprache und GML basierend auf XML Schema.
- INTERLIS mit drei unterschiedlichen, allerdings nicht gleichwertigen Repräsentationen (UML, INTERLIS-Beschreibungssprache und XML Schema) des Datenmodells im Gegensatz zu GML mit XML Schema als einziger offizieller Repräsentation.
- Eine Unterschiedliche Handhabung von Modellspezifikation und Codierungsregeln mit einer weitgehenden Trennung im Falle von INTERLIS (Modellbeschreibung in INTERLIS und Beschreibung des Datentransfers in XML Schema) und einer kombinierten Lösung in GML (Modell- und Transferbeschreibung in XML Schema).
- Eine praxiserprobte Lösung im Falle von INTERLIS mit einem verkraftbaren Normierungstempo.
- INTERLIS ist für Menschen verhältnismässig einfach zu lesen, GML (bzw. XML Schema) ist ohne Hilfsmittel nur sehr schwierig interpretierbar.

6.5. Fazit

INTERLIS 2 und GML 3 sind zwei durchaus gleichwertige, wenn auch andersartige Mechanismen für die Beschreibung und den modell-basierten Austausch von Geodaten. INTERLIS hat seine Stärken in einer eleganten Modellierungssprache und einer klaren Trennung zwischen Modellbeschreibung und Codierung des Transfers. GML weist demgegenüber eine grössere Mächtigkeit auf und ist in einen gewichtigen Normierungsprozess eingebunden, der ein breites Spektrum an geo-relevanten Fragestellungen abdeckt. Ein deutlicher Unterschied besteht in der Bewährung in der Praxis, welche bei GML zurzeit noch aussteht.

INTERLIS 2 und GML 3 weisen ein hohes Mass an Kompatibilität auf, d.h. Datenmodelle und Datensätze in INTERLIS können weitgehend verlustfrei in GML 3 übertragen werden. Für die Realisierung dieser Transformation bieten sich im heutigen Zeitpunkt die Varianten 3 (Abbildung zwischen den beiden normativen Beschreibungen) und 3a (modell-basierte XSL Transformation) an (vgl. 5.4.3). Die Umkehrung, insbesondere für beliebige GML-Datenmodelle, dürfte sich auf Grund der grösseren Mächtigkeit von GML 3 (3D, Coverages, Topologie) deutlich schwieriger gestalten und in vielen Fällen nur mit Informationsverlust möglich sein.

Auf Grund der aktuellen Ausgangslage drängt sich eine Ablösung von INTERLIS durch GML kurz- bis mittelfristig nicht auf. Hingegen dürfte eine kombinierte Lösung, konkret die Ergänzung von INTERLIS um eine GML-Transferoption, für bestehende INTERLIS-AnwenderInnen schon sehr bald einen beträchtlichen Nutzen bieten (vgl. auch 2.1 zur Bedeutung von Geonormen).

Weiteres Vorgehen – Szenarien und

Empfehlungen

Für die zukünftige Strategie der Geonormierung in der Schweiz bieten sich auf Grund der durchgeführten Untersuchungen und vor dem Hintergrund der weltweiten Aktivitäten rund um GML – mit nach wie vor etwas ungewissem Ausgang – die folgenden 3 Hauptszenarien an:

1. Weiterentwicklung von INTERLIS unabhängig von GML
2. Zweigleisige Strategie mit Konvertierungsschiene
3. Rasche vollständige Umstellung auf GML

Diese Szenarien werden in den folgenden Abschnitten kurz beschrieben und bezüglich Aufwand, Chancen und Risiken bewertet.

7.1. Szenario 1 – Unabhängige Weiterentwicklung von INTERLIS

In diesem Szenario würde INTERLIS beibehalten und konsequent weiter entwickelt, möglicherweise mit der Spezifikation von darauf aufbauenden Diensten. Die GML-Entwicklung würde weiter beobachtet, ohne aktiv Einfluss darauf zu nehmen. Die Entwicklung von Konvertierungstools für Datenmodelle oder sogar Daten zwischen INTERLIS und GML würde von offizieller Seite nicht gefördert. Nach diesem Szenario wäre INTERLIS in der Schweiz in nächster Zeit der einzige offizielle Datenmodellierungs- und Transfermechanismus.

Der Aufwand für dieses Szenario wäre kurz- bis mittelfristig gering.

Chancen

Es besteht die wohl geringe Möglichkeit eines Scheiterns von GML wegen zu grosser Komplexität, zu raschem Entwicklungstempo etc. In diesem sehr unwahrscheinlichen Fall, hätte man mit INTERLIS eine funktionierende Alternative anzubieten.

Risiken

Dieses Szenario birgt die folgenden Risiken:

- Schweizerisches Know-how im Bereich der modellbasierten Modellierung würde nicht oder nur beschränkt in die Weiterentwicklung von GML einfließen.
- Die Schweiz könnte die Definition von internationalen Datenmodellen bzw. internationalen GML-Applikationsschemas nicht beeinflussen.
- Die Realisierung von Geoinformationsdiensten geschieht bereits heute primär auf der Basis von OGC-Spezifikationen (WMS und WFS). Ohne geeignete ILI-GML-Konvertierungstools ist dabei eine Nutzung bestehender INTERLIS-Modelle und –Datensätze in Frage gestellt.
- Allfällige zukünftige Geoinformationsdienste auf INTERLIS-Basis wären nur lokal nutzbar (ILI-Territorium) und dürften daher auf dem Markt einen sehr schweren Stand haben.

7.2. Szenario 2 – Zweigleisige Strategie mit Konvertierungsschiene

Dieses Szenario könnte auch betitelt werden als 'INTERLIS-Modellierung mit GML-Unterstützung'.

Beschreibung

In diesem Szenario wird eine Brücke zwischen INTERLIS und GML geschlagen, um die Voraussetzungen für einen Parallelbetrieb oder eine allfällige zukünftige Umstellung zu schaffen. Die möglichen Massnahmen zur Umsetzung dieses Szenarios umfassen:

- Vertiefung der Untersuchungen von GML 3.
- Die Definition der Abbildungsregeln zwischen INTERLIS und GML.
- Entwicklung von Tools zur Generierung von GML-Applikationsschemas aus INTERLIS-Modellen und zur Konvertierung von INTERLIS-Datensätzen in GML. Dies würde idealerweise durch eine Erweiterung des INTERLIS-Compilers geschehen.
- Konvertierung von wichtigen Datenmodellen in GML (z.B. AV01).
- Einbringen des CH-Know-hows in den Normierungsprozess und aktive Einflussnahme auf die Weiterentwicklung von GML.
- Reduktion der INTERLIS-Entwicklung in Richtung Konsolidierung von INTERLIS 2.

Chancen

Die Chancen dieses Szenarios wären:

- Die Möglichkeit eines einfachen und vollständigen Umstiegs auf GML – bei einer Konsolidierung des Standards – oder einer vollständigen Beibehaltung von INTERLIS für den Fall eines Scheiterns von GML.
- Die Möglichkeit einer aktiven Beeinflussung der Weiterentwicklung von GML, vor allem in Richtung model-driven Geodatenmodellierung und –transfer.
- Ermöglicht 'Abwarten' der weiteren Entwicklung von GML und einen definitiven Umstieg bei Stabilisierung der GML-Normierung (ISO-Standard?)
- Die Möglichkeit der Beibehaltung von INTERLIS als Datenbeschreibungssprache in einem GML-Umfeld.
- Eine realistische Möglichkeit zur Übernahme einer führenden Rolle der Schweiz
 - in der Definition von Datenmodellen auf GML-Basis (aus INTERLIS-Konvertierung)
 - in der Entwicklung von Tools zur modell-getriebenen Generierung von GML-Applikationsschemas

Risiken

Die zweigleisige Strategie birgt die Gefahr von aufwändigen Parallelentwicklungen und erfordert daher klare Umsetzungsvorgaben.

7.3. Szenario 3 – Rasche Umstellung auf GML

In diesem Szenario würde die Umstellung auf GML so rasch als möglich erfolgen. Mit der raschen Entwicklung von Schema- und Datenkonvertierungstools könnten alle bestehenden INTERLIS-Datenmodelle und –daten in GML 3 konvertiert werden. Um zukünftig eine effiziente Arbeit mit GML zu ermöglichen müssten ebenfalls rasch geeignete GML-Modellierungstools entwickelt oder evaluiert werden.

Chancen

Durch ein klares 'Commitment' zu GML könnten sich Systemhersteller auf die Entwicklung und Anpassung von GML-Tools konzentrieren.

Risiken

Dieses Szenario birgt die folgenden Risiken:

- Ein 'Übungsabbruch' bei einem Misserfolg von GML, d.h. ein Rückkehr zu INTERLIS, wäre in diesem Szenario relativ schwierig.
- Gefahr laufender Veränderungen im GML-Normierungsprozess (welche GML-Version ist massgebend GML 3.0, 3.1. oder 4.0?).
- Das mit INTERLIS erarbeitete Know-how ginge verloren, bzw. müsste in GML unter sehr grossem Aufwand teilweise neu aufgebaut werden.

7.4. Empfehlungen und Ausblick

Nach Ansicht der Autoren ist Szenario 2 zu favorisieren. Dieses Szenario bietet einen raschen, konkreten Anwendernutzen bei vernünftigem technischem Aufwand und bei relativ geringem Risiko. Mit Szenario 2 können die Chancen des neuen internationalen Standards GML wahrgenommen werden, ohne das Bewährte über Bord werfen zu müssen. Dabei können insbesondere die Vorzüge beider Mechanismen kombiniert und bestehendes Know-how eingebracht werden. Szenario 2 erlaubt es auch, den weiteren Verlauf der GML-Normierung mit zu verfolgen und möglicherweise zu beeinflussen, ohne Gefahr zu laufen, eine verfrühte Umstellung auf einen noch unreifen Standard zu vollziehen.

Konkret werden dazu die folgenden Massnahmen und Umsetzungsschritte vorgeschlagen:

- Der INTERLIS-Compiler sollte möglichst rasch um die folgenden zwei Optionen erweitert werden: Eine Funktion zur Erzeugung eines GML-Applikationsschemas aus einer INTERLIS-Modellbeschreibung (Variante 3, Abbildung 5 und Kap. 5.4.3) und eine Funktion zur Erzeugung eines XSL Style Sheets (Variante 3a) mit den Instruktionen zur Konvertierung der eigentlichen INTERLIS-Datensätze in GML.

- Der Quellcode dieser GML-Generatoren sollte zusammen mit dem demjenigen für den bestehenden 'INTERLIS-Parser' des INTERLIS-Compilers gut dokumentiert und frei zur Verfügung gestellt werden, um eine breite Umsetzung zu fördern.
- Bei Vorliegen des neuen INTERLIS-Compilers sollten für die bestehenden INTERLIS-Datenmodelle möglichst umgehend die entsprechenden GML-Applikationsschemas und XSL Style Sheets erzeugt und publiziert werden. Damit würde die Schweiz sehr rasch über einen umfassenden Fundus von domänenspezifischen Datenmodellen in GML verfügen.

Die Erfahrungen aus diesen ersten Umsetzungsschritten und aus den begleitenden Untersuchungen dürften eine gute Basis bilden für eine praxisorientierte Weiterentwicklung der Geonormen-Strategie in der Schweiz.

Mit den oben aufgeführten ersten Umsetzungsschritten wäre auch die Voraussetzung geschaffen, um bestehende INTERLIS-Daten in zukünftigen Geodatendiensten in einem OGC-Kontext zu nutzen – entweder via vorgängige oder on-the-fly Transformation.

Mit Szenario 2 könnten ausserdem wichtige Vorarbeiten geleistet und Erfahrungen gesammelt werden für einen möglichen zukünftigen Übergang auf eine Modelltransformation via UML und XMI. Dazu müssten aber vorgängig eine Reihe von Voraussetzungen geschaffen werden, beispielsweise die verbindliche Festlegung von Abbildungsregeln zwischen UML und XML Schema auf der Seite von GML.

8. Literatur / Referenzen

8.1. Literatur

Academia Sinica Computing Centre, T., 2003. The Schematron - An XML Structure Validation Language using Patterns in Trees,

<http://www.ascc.net/xml/resource/schematron/schematron.html>.

Andresen, A., 2003. Komponentenbasierte Softwareentwicklung mit MDA, UML und XML. Hanser, München, XIV, 353 S. pp.

Bernard, L. und Streit, U., 2002. Geodateninfrastrukturen und Geoinformationsdienste: Aktueller Stand und Forschungsprobleme, Publikationen der Deutschen Gesellschaft für Photogrammetrie und Fernerkundung (Band 11).

Dorfschmid, J. und Brawer, S., 2003. Modellieren raumbezogener Daten - Eine Einführung unter Berücksichtigung von UML und INTERLIS. KOGIS/COSIG,

http://www.interlis.ch/interlis2/download_d.php.

Frankel, D.S., 2003. Model driven architecture applying MDA to enterprise computing. Wiley, New York, 328 S. pp.

Galdos Systems Inc., 2002. Mineral Occurrence Application Schema, TR2001-213-03. Submitted to Natural Resources Canada,

http://cqkn.net/2002/projects/xml/docs/mineral_occurance_schema.pdf.

KOGIS, 2003. INTERLIS 2 - Referenzhandbuch,

http://www.interlis.ch/interlis2/download_d.php.

OMG Object Management Group, 2002. Meta Object Facility (MOF) Specification,

<http://www.omg.org/docs/formal/02-04-03.pdf>.

OMG Object Management Group, 2003. OMG Unified Modeling Language Specification,

<http://www.omg.org/docs/formal/03-03-01.pdf>.

OMG Object Management Group, 2004. OMG Model Driven Architecture,

<http://www.omg.org/mda>.

Open GIS Consortium, 2003a. OpenGIS Geography Markup Language (GML) 3.0 Implementation Specification, <http://www.opengis.org/docs/02-023r4.pdf>.

Open GIS Consortium, 2004a. OpenGIS Geography Markup Language (GML) 3.1 Recommendation Paper, <http://www.opengis.org/specs/?page=recommendation>.

Open GIS Consortium, 2003b. OpenGIS Web Feature Service (WFS) Implementation Specification, V 1.0, OGC 02-058, <http://www.opengis.org/docs/02-058.pdf>.

Open GIS Consortium & ISO/TC211, 2002. The OpenGIS Abstract Specification - Topic 12: OpenGIS Service Architecture; ISO/DIS 19119,

<http://www.opengis.org/docs/02-112.pdf>.

Provost, W., 2003. UML for W3C XML Schema Design. XML.COM,
http://www.xml.com/lpt/a/2002/08/07/wxs_uml.html.

SOGI Fachgruppe GIS-Technologie, 2003. Worin liegt der praktische Nutzen von Interoperabilität und Normung für den GIS-Anwender in der Schweiz?,
<http://www.sogi.ch/sogi/Technologie1.pdf>.

8.2. Online-Ressourcen

W3C-Standards (XML-Schema, XLink, etc)

- W3C (World Wide Web Consortium) allgemein: <http://www.w3.org>
- XML-Schema-Spezifikationen:
XML Schema Part 0: Primer: <http://www.w3.org/TR/xmlschema-0/>
XML Schema Part 1: Structures: <http://www.w3.org/TR/xmlschema-1/>
XML Schema Part 2: Datatypes: <http://www.w3.org/TR/xmlschema-2/>
- XLink-Spezifikation:
XML Linking Language (XLink) Version 1.0: <http://www.w3.org/TR/xlink/>

OMG-Standards (UML, MOF, XMI, etc.)

- Website von Prof. Mario Jeckle mit umfangreichen Information und Linksammlungen zu UML, XML, XMI und Web Services www.jeckle.de

GML – Projekte, Applikationsschemas und Anwendungen

- Beispiel eines WMS-Clients, welcher auf GML beruht
<http://aries.geo.census.gov/WebTIGER/index.html>
- Aktivitäten bei der Arbeitsgemeinschaft der Vermessungsverwaltungen der Bundesrepublik Deutschland (AdV), insbesondere das Projekt NAS (Normenbasierte Austauschchnittstelle), welches auf GML 3.0 beruht: <http://www.adv-online.de/extdeu/broker.jsp?uMen=bb2708b9-4c8a-3bff-de23-50376a112976>

9. Anhang

9.1. Anhang 1 – Typzuordnung zwischen INTERLIS 2 und GML 3

BasisTyp INTERLIS	Beispiel für INTERLIS	INTERLIS-Werte- bereich	empfohlene GML-Abbildung	GML-Wertebereich	Bemerkungen
TextType	TEXT [*Max-Length]	UTF-8	xsd:string [mit restrictions]	siehe xml-header	
	NAME	Name = Letter { Letter Digit ' _ ' }	gml:name	xsd:string	Einschränkung von Sonderzeichen geht verloren, wenn keine zusätz- lichen Einschränkungen definiert werden
	URI	gemäss Spezifikation w3.org	xlink:href		
EnumerationType	Aufzählungen (einfach)	TEXT	xsd:enumeration	xsd:string	
	Aufzählungen (geschachtelt)	TEXT	xsd:union		
AlignmentType	HALIGNMENT	Aufzählung (einfach)	xsd:enumeration		
	VALIGNMENT	Aufzählung (einfach)	xsd:enumeration		
BooleanType	BOOLEAN	true, false	xsd:boolean	true, false	
NumericType	NUMERIC	Minimal- und Maximalwert	xsd:decimal mit restrictions		Einschränkung Nachkommastellen über Patterns (Anwendungs- muster) möglich
StructuredUnitType	Bsp: UhrZeit	Minimal- und Maximalwert, Verweis auf Referenzunit	xsd:time		
	Bsp: Datum	Minimal- und Maximalwert, Verweis auf Referenzunit	xsd:date		GML: nur englisches Datums- format möglich
CoordinateType	COORD	basiert auf NUMERIC, Verweis auf Referenz- system bei Ein- schränkungen (Bsp. LKoord)	gml:pos	basiert auf xsd:string, Ver- weis auf Referenzsystem durch Attribut srs	Einschränkungen sind komplex, Beschränkung Nachkommastellen nicht möglich

BasisTyp INTERLIS	Beispiel für INTERLIS	INTERLIS-Werte- bereich	empfohlene GML-Abbildung	GML-Wertebereich	Bemerkungen
OIDType	ANYOID (abstract)		gml:id	xsd:string	
	I32OID	positive in 4Bytes spei- cherbare Integerwerte	gml:id	xsd:string	
	STANDARDROID	TEXT*16	gml:id	xsd:string	
BasketType	BASKET	DATA, VIEW, BASE und/oder GRAPHIC	gml:featureCollections	Einschränkungen möglich	
ClassType	STRUCTURE	ähnlich wie Klasse, enthält aber keine Objekte	wie Klasse		
	CLASS		erweiterung vom Klassen- typ (xsd:extension oder xsd:restriction)		
Linienzüge	POLYLINE	basiert auf COORD	gml:CurveArray	basiert auf gml:pos	nicht LKoord, bei ARCS bzw. gml:ArcString können in GML nur Bögen mit drei Punkten gespei- chert werden, Einschränkung nur STRAIGHTS, ARCS geht verloren; keine Overlaps modellierbar
	DIRECTED POLYLINE	basiert auf COORD	gml:OrientableCurve	basiert auf gml:pos	
Einzelflächen und Gebietseinteilungen	SURFACE	basiert auf COORD	gml:Polygon	basiert auf gml:Curve	nicht LKoord, bei ARCS bzw. gml:ArcString können in GML nur Bögen mit drei Punkten gespei- chert werden, Einschränkung nur STRAIGHTS, ARCS geht verloren; keine Overlaps modellierbar
	AREA	basiert auf COORD	gml:Surface	basiert auf gml:Polygon	dito SURFACE, zudem unter- schiedliche Geltungsbereiche (ILI = Klasse; GML = Objekt)

9.2. Anhang 2 – Beispiel AREA in GML

Das untenstehende Beispiel soll eine mögliche Abbildung des INTERLIS-Datentypes AREA in GML 3.0 näher bringen. Das dargestellte GML-Applikationsschema stellt eine gültige Modellbeschreibung gemäss Spezifikation dar. Im untenstehenden Instanzdokument ist ein minimaler Datensatz mit zwei Flächen dargestellt, welche gemäss GML-Spezifikation zu einer Surface (d.h. homogene Region einer Ebene) zusammengestellt sind. Dies in INTERLIS sehr einfach zu beschreibende Modell, führt auf Grund der verschiedenen Konsistenzbedingungen zum Datentyp AREA in GML zu komplexen Konstrukten. Deshalb ist zu beachten, dass aus diesem Beispiel **kein Vergleich** von INTERLIS und GML gezogen werden kann. Zum besseren Verständnis der beiden XML-Dokumente wird empfohlen den unten aufgeführten Quelltext in einen XML-Editor (z.Bsp. XML-Spy) zu kopieren, und die entsprechenden Pfade zu den GML-Basischemen (Version 3.0.1) im Applikationsschema manuell anzupassen.

Beispiel Modellierung in INTERLIS

```
INTERLIS 2.2;
MODEL Beispiel (de) =

  TOPIC BeispielAREA =
    DOMAIN
      LKoord= COORD NUMERIC [INTERLIS.m], NUMERIC [INTERLIS.m];

    CLASS Flaechenelement =
      beschr : MANDATORY TEXT;
      geometrie : MANDATORY AREA WITH (ARCS,STRAIGHTS) VERTEX LKoord;
    END Flaechenelement;

  END BeispielAREA;
END Beispiel.
```

Darstellung in GML-Applikationsschema (beispielArea.xsd)

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- edited with XML Spy v4.3 U (http://www.xmlspy.com) by fhbb (Fachhochschule bei der
Basel) -->
<xsd:schema targetNamespace="http://www.fhbb.ch/ILIttoGML"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ili="http://www.fhbb.ch/ILIttoGML" xmlns:gml="http://www.opengis.net/gml"
elementFormDefault="qualified" version="3.0.1">
  <xsd:import namespace="http://www.opengis.net/gml"
schemaLocation="..\..\GMLSchemas\Basic\gml.xsd"/>
<!-- =====
      TOPIC:  BeispielAREA
===== -->
<xsd:element name="Model" type="gml:AbstractFeatureCollectionType"/>
<xsd:element name="modelMember" type="ili:ModelMemberType"
substitutionGroup="gml:featureMember"/>
<xsd:element name="_ModelFeatureCollection" type="gml:AbstractFeatureCollectionType"
abstract="true" substitutionGroup="gml:_FeatureCollection"/>
<xsd:complexType name="ModelMemberType">
  <xsd:complexContent>
    <xsd:restriction base="gml:FeaturePropertyType">
      <xsd:sequence>
        <xsd:element ref="ili:_ModelFeatureCollection" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="BeispielAREA" type="gml:AbstractFeatureCollectionType"
substitutionGroup="ili:_ModelFeatureCollection"/>
<xsd:element name="bspMember" type="ili:BspMemberType"
substitutionGroup="gml:FeatureMember"/>
<xsd:complexType name="BspMemberType">
  <xsd:complexContent>
    <xsd:restriction base="gml:FeaturePropertyType">
      <xsd:sequence>
        <xsd:element ref="ili:_BspFeature" minOccurs="0"/>
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="_BspFeature" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="gml:_Feature"/>
<!-- ~~~~~~
      Class:  AREAFlaechenelement
~~~~~ -->
<xsd:element name="AREAFlaechenelement" type="gml:SurfaceType"
substitutionGroup="ili:_BspFeature"/>
<!-- ~~~~~~
      Class:  Flaechenelement
~~~~~ -->
<xsd:element name="Flaechenelement" type="ili:FlaechenelementType"
substitutionGroup="ili:_BspFeature"/>
<xsd:complexType name="FlaechenelementType">
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractFeatureType">
      <xsd:sequence>
        <xsd:element name="beschr" type="xsd:string"/>
        <xsd:element name="geometrie" type="gml:SurfacePropertyType"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```

Darstellung in XML-Instanzdokument (beispielArea.gml)

```

<?xml version="1.0" encoding="UTF-8"?>
<BeispielAREA xsi:schemaLocation="http://www.fhbb.ch/ILIttoGML beispielAREA.xsd"
xmlns="http://www.fhbb.ch/ILIttoGML" xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xlink="http://www.w3.org/1999/xlink">
  <gml:boundedBy>
    <gml:Null>unknown</gml:Null>
  </gml:boundedBy>
  <bspMember>
    <AREAFlaechenelement>
      <gml:patches>
        <gml:PolygonPatch>
          <gml:exterior>
            <gml:Ring gml:id="element1">
              <gml:curveMember>
                <gml:LineString>
                  <gml:pos>699840.474 262183.734</gml:pos>
                  <gml:pos>699842.919 262185.992</gml:pos>
                  <gml:pos>699843.069 262185.851</gml:pos>
                  <gml:pos>699844.533 262184.481</gml:pos>
                  <gml:pos>699844.675 262184.348</gml:pos>
                  <gml:pos>699841.629 262181.902</gml:pos>
                  <gml:pos>699840.474 262183.734</gml:pos>
                </gml:LineString>
              </gml:curveMember>
            </gml:Ring>
          </gml:exterior>
        </gml:PolygonPatch>
        <gml:PolygonPatch>
          <gml:exterior>
            <gml:Ring gml:id="element2">
              <gml:curveMember>
                <gml:LineString>
                  <gml:pos>700017.049 261976.683</gml:pos>
                  <gml:pos>700021.794 261981.884</gml:pos>
                  <gml:pos>700023.293 261980.516</gml:pos>
                  <gml:pos>700018.575 261975.290</gml:pos>
                  <gml:pos>700017.049 261976.683</gml:pos>
                </gml:LineString>
              </gml:curveMember>
            </gml:Ring>
          </gml:exterior>
        </gml:PolygonPatch>
      </gml:patches>
    </AREAFlaechenelement>
  </bspMember>
  <bspMember>
    <Flaechenelement>
      <beschr>erstes Flaechenelement</beschr>
      <geometrie xlink:href="#element1"/>
    </Flaechenelement>
  </bspMember>
  <bspMember>
    <Flaechenelement>
      <beschr>zweites Flaechenelement</beschr>
      <geometrie xlink:href="#element2"/>
    </Flaechenelement>
  </bspMember>
</BeispielAREA>

```

Flächen werden als PolygonPatches "global" deklariert. Flächen innerhalb dieses FeatureMembers müssen homogen sein (siehe Datentyp surface [GML-Spezifikation])

mittels xlink-Attributen werden die Teilflächen referenziert