

Webbasierte Modelltransformation in der Geoinformatik

Andreas Donaubaue¹, Tatjana Kutzner², Hans Rudolf Gnägi³, Stefan Henrich⁴ und Astrid Fichtinger⁵

Abstract: In der Geoinformatik, welche sich mit der Modellierung raumbezogener Strukturen und Prozesse in Geoinformationssystemen (GIS) beschäftigt, spielt die Modelltransformation eine große Rolle. Neu ist dabei aus Sicht der Geoinformatik, dass die Modelltransformation nicht auf Format-/Datenebene durch bilaterales Umformatieren implementiert wird, sondern auf die Ebene der konzeptionellen Modelle (UML) gehoben wird. Dort kann die Modelltransformation anschaulich grafisch definiert werden und die (physische) Datentransformation aufgrund der konzeptionell definierten Modelltransformation anschließend automatisch erfolgen. In diesem Beitrag wird eine Methode vorgestellt, wie sowohl Modell- als auch Datentransformation webbasiert durchgeführt werden können. Die Methode besteht aus einer Sprache zur Beschreibung von Transformationsregeln (UMLT), einer Webschnittstelle für den Aufruf von Modelltransformationen (mdWFS) sowie einem Zugriff auf die transformierten Geodaten innerhalb einer serviceorientierten Architektur. Die prototypische Implementierung dieses webbasierten Modell- und Datentransformationssdienstes erfolgte vor dem Hintergrund der EU-Richtlinie INSPIRE, welche einen europaweit einheitlichen Zugriff auf die Geodaten der Mitgliedstaaten erfordert. Die positiven Ergebnisse wie auch beim Prototyping erkannte Probleme werden im letzten Abschnitt erläutert. Sie sind z.T. Themen für weitergehende Forschung.

1 Einführung

Die Geoinformatik beschäftigt sich mit der Modellierung von Strukturen und Prozessen mit Raumbezug. Wesentlicher Nutzen von Geoinformationssystemen (GIS) ist es, Daten aus unterschiedlichsten Quellen aufgrund ihres Raumbezugs zusammenzuführen, um daraus neue Informationen zu gewinnen. Das Zusammenführen von Daten unterschiedlicher Quellsysteme in ein Zielsystem erfordert den Transfer von Daten. Dieser Datentransfer ist jedoch aufgrund der charakteristischen Eigenschaften von Geodaten (Geometrie als Nicht-Standard-Datentyp, Mehrfachrepräsentation von Objekten in unterschiedlichen Detaillierungsgraden, Unschärfe infolge nicht klar abgrenzbarer räumlicher Phänomene, etc.) sowie aufgrund heterogener und gewachsener Strukturen bei den Anbietern raumbezogener Daten nicht trivial. Beim Datentransfer ist deshalb oft notwendig, zusätzlich eine Modelltransformation durchzuführen.

¹ Institut für Geodäsie und Photogrammetrie, Gruppe Geoinformationssysteme, ETH Zürich, Wolfgang-Pauli-Strasse 15, CH-8093 Zürich, donaubauer@geod.baug.ethz.ch

² Fachgebiet Geoinformationssysteme, Technische Universität München, Arcisstr. 21, D-80333 München, tatjana.kutzner@bv.tum.de

³ Institut für Geodäsie und Photogrammetrie, Gruppe Geoinformationssysteme, ETH Zürich, Wolfgang-Pauli-Strasse 15, CH-8093 Zürich, gnaegi@geod.baug.ethz.ch

⁴ Institut für Geodäsie und Photogrammetrie, Gruppe Geoinformationssysteme, ETH Zürich, Wolfgang-Pauli-Strasse 15, CH-8093 Zürich, henrich@geod.baug.ethz.ch

⁵ Fachgebiet Geoinformationssysteme, Technische Universität München, Arcisstr. 21, D-80333 München, astrid.fichtinger@bv.tum.de

Im Rahmen so genannter Geodateninfrastrukturen sollen Daten und Zugriffsdienste so harmonisiert werden, dass das Zusammenführen der Daten auf einfache Weise möglich ist. Die INSPIRE-Richtlinie der Europäischen Union [EU07] zum Aufbau einer europaweiten Geodateninfrastruktur verlangt von den Mitgliedstaaten, Geodaten über interoperable Geodienste bereitzustellen. Hierfür werden in den INSPIRE-Durchführungsbestimmungen standardisierte Schnittstellen erarbeitet, die einen einheitlichen Zugriff auf die Geodienste aller Mitgliedstaaten ermöglichen und die Daten in einheitlichen Datentransferformaten bereitstellen sollen. Damit für diese Daten auch auf semantischer Ebene, also auf Ebene der Datenmodelle Interoperabilität gegeben ist, werden im Rahmen der INSPIRE-Durchführungsbestimmungen zudem harmonisierte, europaweit einheitliche Datenmodelle (INSPIRE Data Specifications) für bestimmte Themen erstellt, z.B. für Verwaltungsgrenzen oder für Schutzgebiete. Aus Sicht der Anbieter derartiger Daten (z.B. Vermessungsverwaltungen der EU-Mitgliedstaaten) bedeutet dies, dass eine Transformation der Daten aus ihren eigenen Datenmodellen in die von der EU vorgegebenen Modelle durchzuführen ist. Die INSPIRE-Richtlinie sieht für diese Aufgabe sogenannte Transformationsdienste vor. Für derartige Dienste gibt es jedoch noch keine gültigen Standards [Mü08] und es herrscht Forschungsbedarf in diesem Bereich (vgl. z.B. [DSS07], [SGM08], [Le07], [Ba07], [OC08]). Genau diese Lücke schließt die in diesem Beitrag vorgestellte Methode zur webbasierten Modelltransformation.

2 Modellbasierter Datentransfer in der Geoinformatik

Die hier beschriebene Methode zur Modelltransformation stützt sich auf das modellbasierte Vorgehen für den Datentransfer, dessen Grundsatz und Anwendung im Folgenden erörtert werden.

Das modellbasierte Vorgehen erweist sich in der Geoinformatik als die zweckmäßigste Möglichkeit, Geodaten zwischen verschiedenen strukturierten Systemen bzw. Schnittstellen auszutauschen. Grundsatz und Vorteile des modellbasierten Vorgehens in diesem Bereich werden deutlich beim Vergleich mit anderen Möglichkeiten.

Die naheliegende Lösung ist, ein Programm zu schreiben, welches das Austauschformat des Quellsystems in das Austauschformat des Zielsystems transformiert. Gemäß Abbildung 1 begibt man sich damit auf die Ebene der Instanzen und programmiert von Hand eine *Instance Translation*. Wenn diese bilaterale Transformationsmethode aber auf n verschiedene Systeme angewendet wird, dann benötigt jedes dieser Systeme $n - 1$ verschiedene Export-Prozessoren. Für $n(n - 1)$ Programme ist jedoch der Entwicklungs- und Wartungsaufwand unverhältnismäßig groß.

Dieser Aufwand kann reduziert werden, wenn als Austauschformat ein Standardformat verwendet wird. Dann benötigt jedes System nur noch einen Import- und einen Exportprozessor. In der Geoinformatik wird solch ein Standardformat als formatbasierter bzw. formatgesteuerter Datentransferstandard bezeichnet. Wenn sich dieses Format ändert, erhält man einen neuen Standard. Um die Daten zu verstehen, die mit diesem Standardformat transferiert werden, müssen dessen Datenfelder beschrieben werden. Wird hierfür

natürliche Sprache eingesetzt, z.B. Deutsch oder Englisch, so kann dies zu vielen Missverständnissen führen. Die meisten Missverständnisse können jedoch ausgeschlossen werden, wenn die Beschreibung des Formats mit einer formalen Sprache wie einer Programmiersprache durchgeführt wird. Der weitere Vorteil dabei ist, dass diese Beschreibung auch durch einen Computer bearbeitet werden kann. Die Formatbeschreibung entspricht in Abbildung 1 dem *Physical Schema* oder Formatschema. Das Formatschema ist abhängig von einem speziellen Format, d.h. plattform-abhängig, und somit ein plattform-spezifisches Modell (PSM).

Der nächste entscheidende Schritt liegt darin, nicht die Felder des Transferformats mittels formaler Sprache zu beschreiben, sondern die Struktur der zu transferierenden Daten und aus diesem so genannten Datenmodell auf konzeptioneller Ebene die Beschreibung eines entsprechenden Transferformats automatisch herzuleiten. Unterschiedlichste Datenmodelle können so mit derselben Sprache präzise beschrieben und deren entsprechenden Transferformate mit denselben Regeln hergeleitet werden. Dieser modellbasierte Standard (Sprache plus Format-, Herleitungs- oder Encodingregeln) muss auch nicht geändert werden, wenn in einem anderen Anwendungsgebiet eine andere Datenstruktur zu transferieren ist.

Die exakte Beschreibung der Datenstruktur heißt Datenmodell oder konzeptionelles Schema der Daten, *Conceptual Schema* in Abbildung 1. Dieses Datenmodell ist nicht mehr abhängig von einem bestimmten Format, also plattform-unabhängig, ein plattform-unabhängiges Modell (PIM) [OMG03 und OMG05].

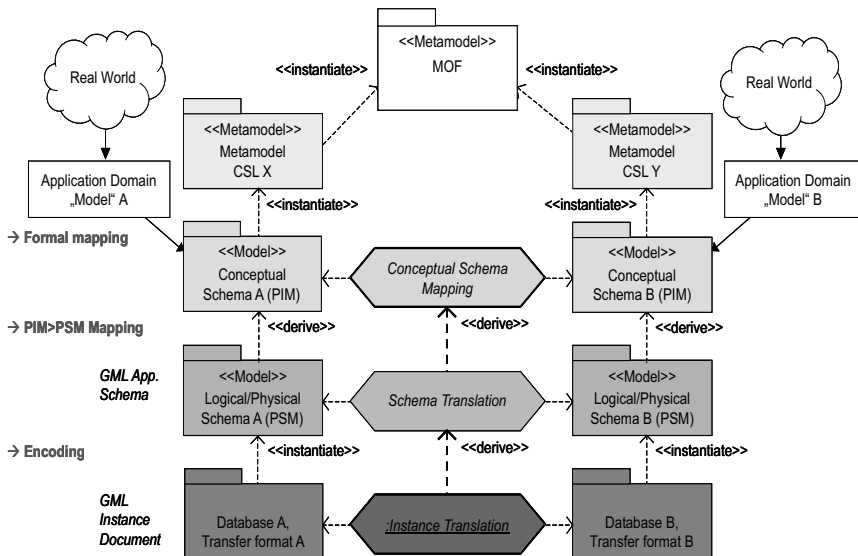


Abb. 1: Das modellbasierte Vorgehen [St07]

Für den modellbasierten Datentransfer muss nun einerseits eine Beschreibungssprache, die sogenannte konzeptionelle Schemasprache (CSL) festgelegt werden (*Metamodel* in Abbildung 1), wovon jedes Datenmodell eine Instanz ist. Andererseits müssen Regeln definiert werden, um die entsprechende Formatbeschreibung aus dem Datenmodell herleiten zu können. Diese Regeln für die Herleitung des Formatschemas sind in Abbildung 1 als *PIM>PSM Mapping* dargestellt.

Dieses modellbasierte Vorgehen dient nicht nur der Standardisierung des Datenaustauschs. Es ermöglicht auch, Werkzeuge zur Verfügung zu stellen für verschiedene Dienste, die aufgrund ihrer Unabhängigkeit von den betreffenden Datenstrukturen systemunabhängig sind. Beispiele für solche Werkzeuge sind:

- a) *Nachhaltige Datensicherung*: Daten, die im Standardformat zusammen mit ihrem Datenmodell abgespeichert sind, bleiben jederzeit interpretierbar, zweifelsfrei und automatisch.
- b) *Geometrische Datentypen*: Unabdingbar für die Modellierung von Anwendungsgebieten der Geoinformatik sind Datentypen für die wesentlichen geometrischen Objekte wie Punkte, Linien, Flächen, Pflasterungen etc. sowie die Möglichkeit, Einheiten und Koordinatensysteme zu definieren. Derartige Datentypen sind in CSL aus der allgemeinen IT wie UML meist nicht vorhanden und werden in CSL-Profilen definiert.
- c) *Automatische Datenprüfung*: Es gibt verschiedene systemunabhängige Checker-Werkzeuge, welche Daten im Standardformat mit dem entsprechenden Datenmodell vergleichen und Inkonsistenzen melden, insbesondere auch geometrischer Art.
- d) *Konzeptionelle Datenbankkonfiguration*: Ein konzeptionell entworfenes Datenmodell kann auf Konsistenz geprüft und unmittelbar in ein konsistentes Datenbankschema übersetzt und damit implementiert werden. Werden Daten aus dieser Datenbank exportiert und wird das Transferformat ebenfalls mittels Regeln aus dem konzeptionellen Schema hergeleitet, so ist sichergestellt, dass die Daten aus der Datenbank im Transferformat verlustfrei abgebildet werden können.
- e) *Modelltransformation (Semantische Transformation)*: Die Datentransformation von einem Quelldatenmodell in ein Zieldatenmodell erfolgt hierbei nicht durch Implementierung auf Datenebene sondern durch Konfiguration einer Modelltransformation auf (konzeptioneller) Modellebene mit Hilfe geeigneter vordefinierter Abbildungselemente.

Während es für die Beispiele a) bis d) bereits Standards und Werkzeuge gibt, müssen im Bereich der Modelltransformation noch grundlegende Forschungsarbeiten geleistet werden. Der Beitrag widmet sich dabei den folgenden Forschungsfragen:

- Wie kann eine Modelltransformation zwischen Datenmodellen für Geodaten auf konzeptioneller Ebene ($PIM \rightarrow PIM$) formal sauber beschrieben werden?
- Gibt es dazu passende $PIM \rightarrow PSM$ -Abbildungen von den Modellen auf die Transferformate und von der Modellabbildung auf die Datentransformation?

- Kann der Datentransfer einschließlich der Modelltransformation als Web Service zur Verfügung gestellt werden?

3 Die Modellabbildungssprache UMLT

Zur Definition von Modelltransformationen auf Ebene der konzeptionellen Datenmodelle wird eine konzeptionelle Abbildungssprache (UMLT genannt) eingeführt. Diese Sprache muss verschiedenen Anforderungen bezüglich ihrer Verwendbarkeit genügen. So dürfen Transformationsschemata nicht nur von Systemen interpretierbar, sondern sie müssen auch für Menschen verständlich sein. Deshalb wurde sowohl ein UML 2-Metamodell als auch eine Syntax für eine *Human Useable Textual Notation* (HUTN) entwickelt. Die Transformationsschemata können dadurch in grafischer Form mit UML-Aktivitätsdiagrammen dargestellt, in textueller Form beschrieben (abgeleitet aus INTERLIS [KG06]) und in XMI (einem XML-basierten Transferformat für Metamodelle) kodiert werden. Einschlägige Normen und Standards der OMG (Object Management Group), des OGC (Open Geospatial Consortium) und der ISO (Internationale Organisation für Normung) wurden bei der Entwicklung berücksichtigt.

3.1 Konzept von UMLT

Bei der Entwicklung der Modellabbildungssprache UMLT wurden insbesondere zwei Formalismen der OMG untersucht. Zum einen die Meta-Object-Facility Query/Views/Transformations-Sprache (MOF-QVT) [OMG05]. Diese Sprache wurde für die Transformation von Metamodellen entwickelt, jedoch sind MOF-QVT-Modelle im Allgemeinen schwer verständlich. Der Standard ist komplex, da eigentlich drei Sprachen spezifiziert werden: Relations, Core und Operational. Darüber hinaus wurde der MOF-QVT-Standard in erster Linie für PIM-PSM-Abbildungen entwickelt und wird bei der Implementierung von Software eingesetzt, wie beispielsweise der Übersetzung von UML nach Java, jedoch weniger für PIM-PIM-Abbildungen.

Der andere untersuchte Formalismus ist das UML 2-Aktivitätsdiagramm. Dieses kann verwendet werden, um Transformationen mittels Aktivitäts-Sequenzen zu beschreiben. Eine klare Beschreibung der Semantik wie auch des Transferformates (XMI 2.1) ist in der Superstructure für UML-Modelle gegeben [OMG07]. UML 2-Modelle sind grundsätzlich leicht verständlich und es sind eine Vielzahl von Implementierungen und Open Source-APIs verfügbar.

Aufgrund dieser Überlegungen wurde das UML 2 Aktivitätsdiagramm als Ausgangspunkt gewählt. Ein wesentlicher Teil der Forschung bestand darin, diese bestehende Technologie um spezielle Abbildungsaktivitäten zu erweitern. Die konzeptionelle Abbildungssprache UMLT ist definiert durch eine unabhängige Erweiterung des UML 2-Metamodells. Unabhängig ist diese Erweiterung in dem Sinne, dass keine Elemente von

UML verändert werden, sondern neue Sprachelemente durch Vererbung hinzugefügt werden (vgl. nachfolgenden Abschnitt).

3.2 UMLT-Sprachelemente

Die Sprachelemente von UMLT sind eine Spezialisierung von UML 2-Aktivitäten [OMG07]. Es werden folgende Sprachelemente eingeführt, welche auch in Abbildung 2 dargestellt sind [St07]. Beispiele für die Verwendung dieser Sprachelemente sind in Kapitel 5 beschrieben:

- a) *TransformationActivity*: Gliederungs- oder Strukturierungselement für semantische Transformationen. Umfasst alle Komponenten für eine gesamte Modelltransformation.
- b) *StructuredTransformation*: Gliederungs- oder Strukturierungselement für semantische Transformationen. Entspricht einer Transformation bestehend aus Objekt- und Kontrollfluss.
- c) *SelectionCriteria*: Auswahl von Input-Daten durch einen logischen Ausdruck.

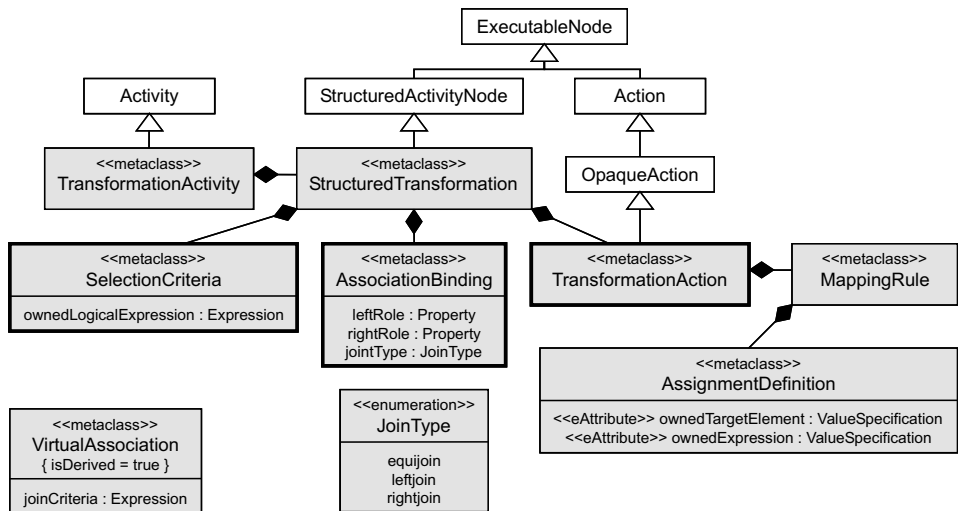


Abb. 2: Sprachelemente des UMLT-Metamodells (weiß: UML 2, grau: UMLT)

- a) *VirtualAssociation*: Verwaltet Input-Daten, die nicht explizit durch eine Assoziation verknüpft sind, aber dennoch einen (engen) Zusammenhang aufweisen. Solche Input-Daten können beispielsweise Referenzattribute oder Fremdschlüsselattribute aufweisen, welche zur Laufzeit ausgewertet werden, um berechnete Relationen zu erhalten. Auf diese Weise können während einer semantischen Transformation solche Objekte virtuell assoziiert werden. Auf welche Weise Objektpaare erzeugt werden, kann mit einem logischen Ausdruck (joinCriteria) explizit defi-

niert werden. Darin unterscheidet sich die virtuelle Assoziation von einer herkömmlichen abgeleiteten Assoziation in UML 2. JoinCriteria können auch auf die geometrischen Attribute einer Klasse angewendet werden. So können z.B. aus einer Klasse „Flüsse“ und einer Klasse „Straßen“, die in einer virtuellen Assoziation über ein Join-Kriterium „crosses“ assoziiert sind, Objektpaare einander kreuzender Flüsse und Straßen ermittelt werden.

- b) *TransformationAction*: Vererbung einer UML OpaqueAction. Bietet ein Aktivitätselement, welches nicht weiter strukturiert werden kann. Eine TransformationAction bildet den Behälter für MappingRules, die auf eine ausgewählte Menge von Objekten angewandt werden sollen.
- c) *AssignmentDefinition*: Spricht Primitivtypen oder Ausdrücke als Wertzuweisung einer TransformationAction an. Dabei kann auch auf Funktionen zurückgegriffen werden. So lassen sich beispielsweise Attributwerte miteinander verknüpfen (siehe Anwendungsfall in Kapitel 5), Value Maps anwenden (z.B. der Wert „18.5“ des Attributs „Wassertemperatur“ im Quellmodell entspricht dem Wert „high“ im Zielmodell), oder Geometriedatentypen ineinander umwandeln (z.B. Typ „Fläche“ in Typ „Linie“).
- d) *MappingRule*: Die eigentliche Objektabbildung. Wird als Komposition von AssignmentDefinitions spezifiziert und bildet eine TransformationAction.
- e) *AssociationBinding*: Bei der Selektion der Input-Objekte kann spezifiziert werden, wie assoziierte Objekte während des Inputs ausgewählt werden sollen.
- f) *JoinType*: Ein Aufzähltyp, um die Verbindungsregel einer AssociationBinding zu spezifizieren.

4 Webschnittstelle für die Modelltransformation

Ein Geo Web Service ist ein Dienst, der Funktionalitäten für die Nutzung von Daten mit Raumbezug, sog. Geodaten, bereitstellt, wie z.B. den Zugriff auf Geodaten sowie deren Verarbeitung und Präsentation, und der mittels Schnittstellen über das Internet ansprechbar ist [Do04]. Geo Web Services ermöglichen Interoperabilität, wofür jedoch standardisierte Schnittstellen zum Zugriff auf Geo Web Services notwendig sind. Diesem Ziel widmet sich das Open Geospatial Consortium (OGC) [OGC09] durch Entwicklung frei zugänglicher Schnittstellenspezifikationen für den Zugriff auf Geo Web Services. Nachfolgend wird zuerst die Spezifikation des OGC Web Feature Service erläutert. Anschließend wird diskutiert, inwieweit dieser Dienst eingesetzt werden kann, um die Modelltransformation mittels einer Webschnittstelle zur Verfügung zu stellen.

4.1 OGC Web Feature Service

Ein OGC Web Feature Service (WFS) [VP05] ermöglicht es einer Clientanwendung, über standardisierte Schnittstellen auf entfernte Geodaten zuzugreifen, siehe Abbildung 3. Die Datenhaltung kann dabei auf beliebige Weise erfolgen, z.B. in Datenbanken, was jedoch gegenüber dem Client verborgen bleibt. Die Kommunikation zwischen Client und WFS findet über HTTP statt durch Senden von Anfragen an den WFS. Die von einem Client angefragten Geodaten werden stets in GML (Geography Markup Language) kodiert als Geoobjekte (Features) an den Client ausgegeben. GML ist ein XML-basiertes Transferformat, welches die Formatbeschreibung, Übertragung und Speicherung geographischer Daten ermöglicht. Im Jahre 2007 ist GML von der Internationalen Organisation für Normung (ISO) als internationale Norm ISO 19136 Geographic information – Geography Markup Language (GML) [ISO07] verabschiedet worden.

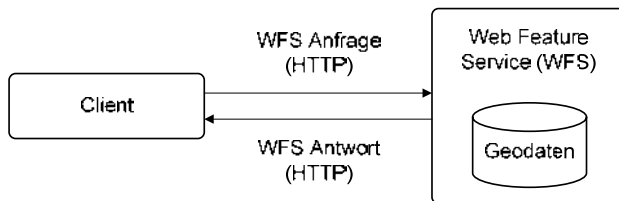


Abb. 3: Web Feature Service (Quelle: [VP05], verändert)

Für den Abruf von Geodaten werden in der WFS-Spezifikation drei Operationen definiert, die jeder WFS verpflichtend bereitstellen muss:

- *GetCapabilities*: Bei Aufruf dieser Operation erhält der Client ein XML-Dokument, in dem die Fähigkeiten des Dienstes durch sogenannte Service-Metadaten beschrieben werden. Zudem wird in diesem Dokument dem Client mitgeteilt, welche Features der Dienst ausgeben kann.
- *DescribeFeatureType*: Mit dieser Operation kann der Client die Formatbeschreibung einzelner Features abrufen.
- *GetFeature*: Über diese Operation werden schließlich die Geodaten als GML-Features in Form einer GML-FeatureCollection an den Client ausgegeben.

Neben dem Abruf von Geodaten (lesender Zugriff) bietet ein WFS auch die Möglichkeit, entfernte Daten zu ändern oder zu löschen (schreibender Zugriff), wofür weitere Operationen zur Verfügung stehen.

4.2 Model Driven Web Feature Service

Der WFS weist jedoch Einschränkungen bezüglich der semantischen Interoperabilität auf. So werden die Daten stets bezogen auf das Datenmodell des Quellsystems geliefert, welches sich i.d.R. vom Modell des Zielsystems unterscheidet. Modellbeschreibungen

auf konzeptioneller Ebene sind in einem WFS – wenn überhaupt vorhanden – für Zielsysteme nicht sichtbar und Modelltransformationen werden nicht unterstützt.

Aus diesem Grund wurde untersucht, inwieweit Modelltransformationen mittels eines Web Service zur Verfügung gestellt werden können. Damit sich solch ein Service bestmöglich in eine OGC Web Services Infrastruktur integrieren lässt, wurde als Grundlage die im vorigen Abschnitt beschriebene OGC-Spezifikation des WFS herangezogen und entsprechend den Anforderungen an Modelltransformationen erweitert. Dieser erweiterte WFS wird im Folgenden als mdWFS (model driven Web Feature Service) bezeichnet. Die Nutzung der mdWFS-Schnittstelle zur Konfiguration einer Modelltransformation und die Nutzung der WFS-Schnittstelle zum Abruf von transformierten Geodaten sind in Abbildung 4 dargestellt und wird im Folgenden näher erläutert.

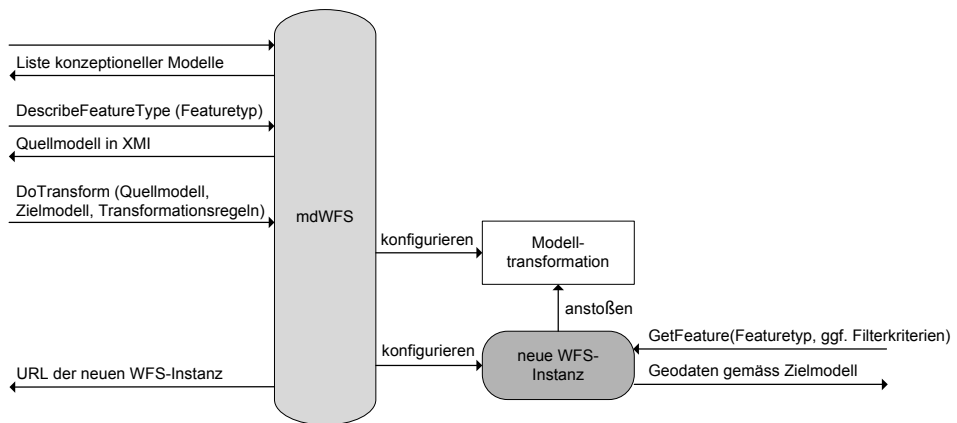


Abb. 4: Nutzung der mdWFS-Schnittstelle

Die WFS-Operation *DescribeFeatureType* wurde für den mdWFS dahingehend erweitert, dass der mdWFS konzeptionelle Datenmodelle im Format XMI bereitstellen kann [St09]. Darüber hinaus wurden für den mdWFS zwei neue Operationen definiert:

- *GetCapabilities*: In Abwandlung der WFS-Spezifikation des OGC enthält die Antwort auf eine *GetCapabilities*-Anfrage keine Liste der verfügbaren Features, sondern eine Liste mit den Namen aller konzeptionellen Modelle, auf die die jeweilige mdWFS-Instanz einen Zugriff erlaubt.
- *DoTransform*: Durch Aufruf dieser Operation werden die in UMLT auf konzeptioneller Ebene beschriebenen Transformationsregeln auf ein plattform-spezifisches Modell übertragen, um letztendlich Daten transformieren zu können. Jeder erfolgreiche Aufruf der Operation resultiert in einer neuen WFS-Instanz, über die die Daten entsprechend dem Zielschema abrufbar sind, siehe Abbildung 4. Dies ist eine grundlegende Designentscheidung, deren großer Vorteil es ist, dass dadurch ein herkömmlicher WFS-Client ohne jegliche Erweiterung für den Zugriff auf die Daten verwendet werden kann. Aufrufparameter der Operation sind: das Quellsche-

ma bzw. die DescribeFeatureType-URL jener WFS-Instanz, die das Quellschema repräsentiert, das Zielschema im Format XMI, die UMLT-Transformationsregeln im Format XMI sowie optional die Wahl von Regeln für das Encoding der Daten.

Die Antwort des mdWFS auf eine DoTransform-Anfrage besteht aus einer Statusmeldung sowie aus einer GetCapabilities-Request-URL als Identifikator für die neu angelegte WFS-Instanz. Ein herkömmlicher WFS-Client kann diese URL nutzen, um auf Funktionalität und Daten dieser WFS-Instanz zuzugreifen. Bei jedem Abruf von Daten über die WFS-GetFeature-Operation wird die Modelltransformation angestoßen und die Daten werden in das Zielmodell transformiert, bevor sie ausgeliefert werden.

5 Anwendungsfall

Zur Demonstration des hier beschriebenen Konzepts wird ein Prototyp entwickelt, welcher eine Modelltransformation zwischen dem deutschen AAA-Modell (Quellschema) und dem von der EU vorgegebenen INSPIRE-Modell (Zielschema) durchführen kann. Als Ausschnitt aus dem Quellmodell wird der Themenbereich *Schutzgebiete* gewählt, welcher in das entsprechende Zielmodell *ProtectedSites* transformiert wird.

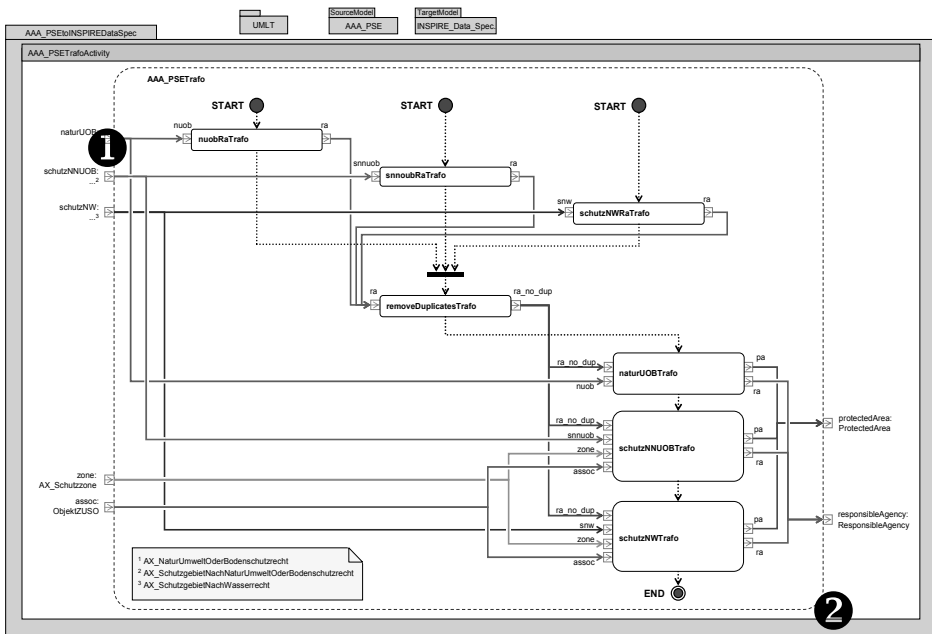


Abb. 5: Darstellung der Modelltransformation in UMLT (Übersicht)

Für die Modelltransformation wurden Transformationsregeln zwischen Quell- und Zielmodell auf konzeptioneller Ebene definiert. Abbildung 5 zeigt in einer Übersicht die Transformation der Schutzgebiete aus dem AAA-Modell in das INSPIRE-Modell unter

Verwendung von UMLT.

Das Package AAA_PSEtoINSPIREDataSpec beinhaltet die gesamte Transformation als *TransformationActivity* AAA_PSETrafoActivity (entspricht dem Punkt a) aus Kapitel 3.2). Zusätzlich wird der Zugriff auf die Packages UMLT, das das Metamodell von UMLT enthält, das Quellmodell und das Zielmodell ermöglicht (Detail ❶).

Innerhalb der TransformationActivity wird die *StructuredTransformation* AAA_PSETrafo aufgebaut (Punkt b) aus Kapitel 3.2), in welcher der Objektfluss von links nach rechts (*Data Flow*, durchgehende Linien) und der Kontrollfluss von oben nach unten (*Control Flow*, gepunktete Linien) festgelegt werden. An den Input-Pins der StructuredTransformation (z.B. naturUOB, schutzNNUOB; siehe Detail ❶) liegen die Objekte aus den benötigten Klassen des Quellmodelles an. Über logische Ausdrücke werden die Objekte gefiltert (*SelectionCriteria*, nicht dargestellt, Punkt c) aus Kapitel 3.2) und den einzelnen *TransformationActions* zugeführt (Punkt e) aus Kapitel 3.2). Der Kontrollfluss steuert dabei die Reihenfolge der auszuführenden TransformationActions.

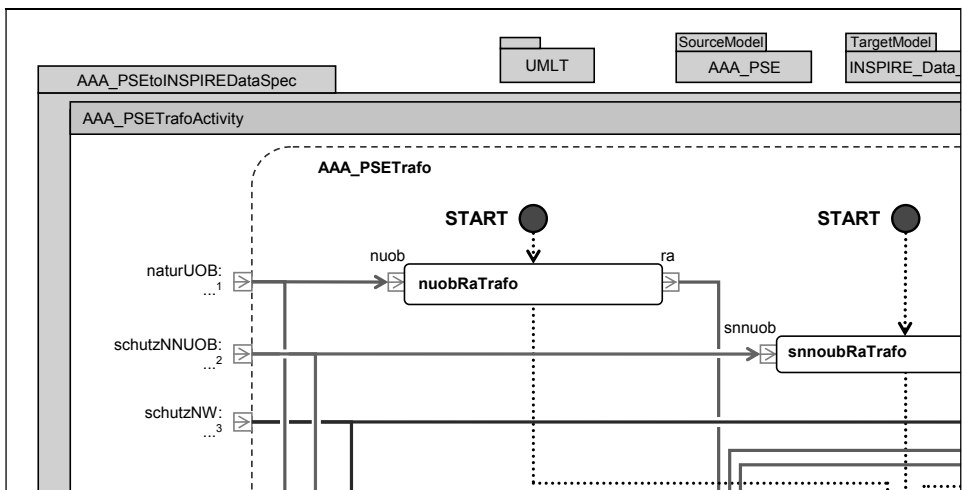


Abb. 6: Detail ❶: Ausschnitt aus der Transformation mit den Startpunkten des Control Flows (START), TransformationActions sowie Input- und Output-Pins des Data Flows

Durch die TransformationAction werden die am Input-Pin anliegenden Objekte sequenziell verarbeitet und anschließend an den Output-Pin weitergeleitet. Die Verarbeitung erfolgt aufgrund der *MappingRules* (Punkt g) aus Kapitel 3.2) für jedes einzelne Objekt. Abbildung 7 zeigt den Einblick in die TransformationAction nuobRaTrafo aus Detail ❶ mit den entsprechenden MappingRules, die ihrerseits aus *AssignmentDefinitions* (Punkt f) aus Kapitel 3.2) aufgebaut sind. Die Notation der MappingRules ist an diejenige von INTERLIS angelehnt. Der Pfeil-Operator „->“ dient der Angabe eines Pfades, welcher zum gewünschten Attribut eines Objektes führt. Lesebeispiel: Die Funktion `String.concatenator` erzeugt einen neuen String aus den zwei Attributwerten

land und stelle des Quellmodells und weist diesen dem Attribut localID aus dem Zielmodell zu.

```

nuobRaTrafo

ra->objectIdentifier->localID := String.concatenator(nuob->
    ausfuehrendeStelle->land, nuob->ausfuehrendeStelle->stelle);

ra->objectIdentifier->namespace := "de";

ra->objectIdentifier->versionId := "Unpopulated";

ra->responsibleAgencyName := "Unpopulated";
    
```

Abb. 7: Eine der sieben TransformationActions des Beispiels

Nach dem Durchlaufen der verschiedenen TransformationActions werden die umgewandelten Objekte am Ende der Transformation entsprechend den Klassen des Zielmodells ausgegeben (Detail ②).

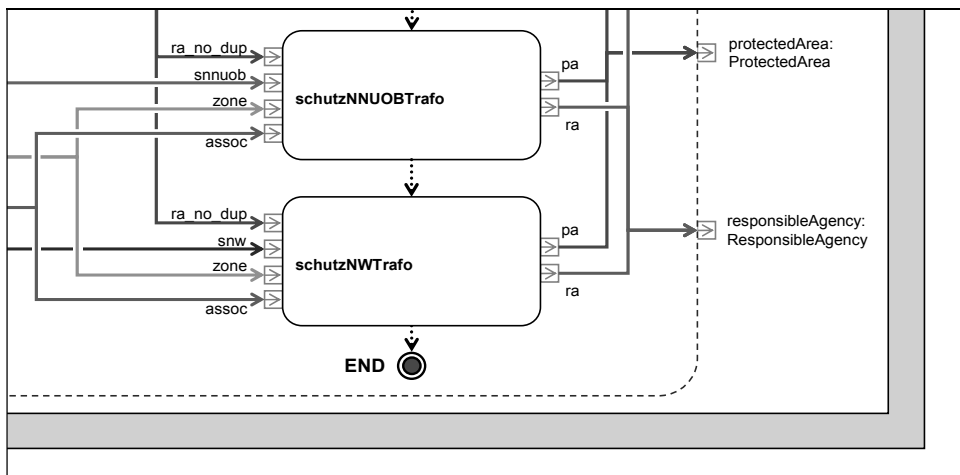


Abb. 8: Detail ②: Ausschnitt aus der Transformation mit dem Endpunkt des Control Flows (END) sowie den Output-Pins der gesamten Transformation

Die in UMLT beschriebenen Transformationsregeln sind nun auf ein plattform-spezifisches Modell zu übertragen, um letztendlich Daten transformieren zu können. Wünschenswert ist es, die Transformation zwischen Quell- und Zielmodell *on-the-fly* durchzuführen, d.h., die Transformation wird jedes Mal dann ausgeführt, wenn ein Benutzer Daten über die WFS-Schnittstelle abrufen. Dies hat den Vorteil, dass die transformierten Daten immer aktuell sind, da Kopien der Daten bezogen auf das Zielmodell nicht abgespeichert werden.

Da eine komplette Eigenentwicklung eines solchen Transformationsmoduls sehr aufwändig ist, wurde entschieden, eine bereits vorhandene Softwarelösung in den Prototyp zu integrieren. Hierfür wurde die Spatial ETL Software FME (Feature Manipulation Engine) [FME09] der kanadischen Firma Safe Software Inc. ausgewählt, welche die Möglichkeit bietet, Mappings zwischen Daten auf Ebene des Datenformats zu definieren und entsprechende Transformationen durchzuführen. Diese Software musste jedoch dahingehend erweitert werden, dass sie sowohl konzeptionelle Datenmodelle als auch die in UMLT formulierten Modelltransaktionsregeln lesen und interpretieren kann.

6 Fazit und Ausblick

Für das Schlüsselproblem der – zumindest im Geoinformatikbereich – neuen Methode der Modelltransformation mit Definition der Transformation von Quellmodell nach Zielmodell auf konzeptioneller Ebene sowie der automatisch daraus folgenden Daten-transformation, konnte eine erfolgreiche Lösung vorgestellt werden, die zudem in eine serviceorientierte Architektur integrierbar ist. Diese webbasierte Modelltransformation besteht aus folgenden Komponenten (in der Reihenfolge, in der sie zum Einsatz kommen):

- Web-Schnittstellen für den Zugriff auf die Liste der verfügbaren konzeptionellen Quelldatenmodelle und für die Bereitstellung des vom Nutzer gewünschten Quelldatenmodells. Das Zieldatenmodell ist beim Nutzer vorhanden.
- Sprache (UMLT) zur Definition der Modelltransformation auf konzeptioneller Ebene mit automatisch folgender Datentransformation auf physischer (Format-) Ebene.
- Web-Schnittstelle zur Übergabe von Quellmodell, Zielmodell und Modelltransaktionsregeln sowie Ausführung der Datentransformation und Rückgabe der Web-Adresse eines Dienstes für den Zugriff auf die transformierten Daten entsprechend des Zieldatenmodells.

Alle drei Komponenten wurden erfolgreich in einem Prototyp implementiert und anhand eines Anwendungsfalls evaluiert. Dieses Ergebnis kann als wesentlicher Fortschritt für den Geodatentransfer mit Modelltransformation und dessen Verfügbarkeit im Web gewertet werden.

Die in Kapitel 2 genannten zentralen Forschungsfragen können somit wie folgt beantwortet werden:

1. Es ist möglich, eine Modelltransformation zwischen Datenmodellen für Geodaten auf konzeptioneller Ebene (PIM \rightarrow PIM) formal sauber zu beschreiben. Die hier beschriebene Sprache UMLT wurde durch wesentliche Erweiterung einer vorhandenen Technologie (UML 2 Aktivitätsdiagramm) entwickelt. Für die Evaluierung des Prototyps wurden in UML modellierte Datenmodelle aus Deutschland, der Schweiz und von INSPIRE als Quell- bzw. Zielmodelle bei der Modelltransforma-

tion verwendet. Dabei zeigte sich, dass in allen drei Fällen ein unterschiedliches UML-Profil zum Einsatz kommt. Dies bedeutet, dass nicht nur unterschiedliche Datenstrukturen umzubauen sind (Datentransformation, PIM→PIM-Transformation), sondern auch die verwendeten Datenbeschreibungssprachen verschieden sind und unterschiedliche Metamodelle besitzen. Außerdem zeigte sich, dass die Datenmodelle aus der Praxis zum Teil unnötig komplex sind und Mehrdeutigkeiten enthalten.

2. Die zweite wesentliche Komponente der hier beschriebenen Methode der Modelltransformation ist die PIM→PSM-Abbildung. Sie wird sowohl für den automatischen Übergang von der konzeptionellen Ebene des Datenmodells zur physischen Ebene von Format bzw. Daten als auch von der konzeptionellen Ebene der Modelltransformationsregeln zur physischen Ebene der Datentransformation verwendet. Dazu braucht es eindeutige Kodierungsregeln. Bei der Evaluierung des Prototyps stellte sich heraus, dass für die drei untersuchten Modelle und die zugeordneten Transferformate jeweils andere Codierungsregeln gelten, was für die Umsetzung der Methode ein Problem darstellt. Im Prototyp wurde dieses Problem gelöst, in dem alle Modelle, teilweise durch aufwändige Nachmodellierung, auf das Metamodel der Schweizer Norm INTERLIS [KG06] gebracht wurden. INTERLIS definiert eindeutige PIM→PSM-Abbildungsregeln (Datenmodell→Transferformat). Für die PIM→PSM-Abbildung der Modelltransformationsregeln konnte durch die Implementierung des Prototyps ein Machbarkeitsnachweis erbracht werden. Da diese Regeln von den internen Strukturen der jeweiligen Transformationssoftware abhängig sind, wurde auf sie in diesem Beitrag nicht näher eingegangen.
3. Es ist möglich, Datentransfer einschließlich Modelltransformation als Web Service zur Verfügung zu stellen (vgl. Kapitel 4). Hierzu mussten vorhandene Standards und Technologien (WFS, Spatial ETL Software FME) um wesentliche Komponenten erweitert und die Realisierbarkeit durch Implementierung eines Prototyps (vgl. Kapitel 5) nachgewiesen werden.

Die hier beschriebenen Probleme wurden zum Anlass genommen, 2010 im Rahmen einer Studie zu untersuchen, ob und wie die Modelltransformation zukünftig unabhängig von speziellen UML-Profilen und Kodierungsregeln eingesetzt werden kann.

Dank

Der Dank für die Finanzierung des Forschungsprojekts mdWFS gilt den Auftraggebern – dem Deutschen Bundesamt für Kartographie und Geodäsie (BKG) sowie dem Schweizer Bundesamt für Landestopographie swisstopo.

Literaturverzeichnis

[Adv06] Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik

- Deutschland (AdV) (Hrsg.): Dokumentation zur Modellierung der Geoinformationen des amtlichen Vermessungswesens (GeoInfoDok) - Version 5.1 - 31.03.2006
- [Ba07] Balley, S.: Aide à la restructuration de données géographiques sur le Web. Dissertation an der Université de Marne-la-Vallée, Frankreich, 2007.
- [Do04] Donaubaue, A.: Interoperable Nutzung verteilter Geodatenbanken mittels standardisierter Geo Web Services, Dissertation, Technische Universität München, 2004.
- [DSS07] Donaubaue, A.; Straub, F.; Schilcher, M.: mdWFS: A Concept of Web-enabling Semantic Transformation. Proceedings of the 10th AGILE Conference on Geographic Information Science, Aalborg, 2007.
- [EU07] Europäisches Parlament und Europäischer Rat (Hrsg.): Richtlinie 2007/2/EG des Europäischen Parlaments und des Rates vom 14. März 2007 zur Schaffung einer Geodateninfrastruktur in der Europäischen Gemeinschaft (INSPIRE). Amtsblatt der Europäischen Union Nr. 108/1 vom 25.4.2007.
- [FME09] Safe Software Inc.: FME, <http://www.safe.com/products/overview.php>
- [IDT08] INSPIRE Drafting Team Data Specifications: D2.7: Guidelines for the encoding of spatial data, Version 3.0.
- [ISO07] Norm ISO 19136:2007, Geographic Information – Geography Markup Language (GML), 2007.
- [KG06] KOGIS(Hrsg.): INTERLIS Referenzhandbuch, Ausgabe vom 13.04.2006 (deutsch). Bundesamt für Landestopographie. Wabern, Schweiz.
- [Le07] Lehto, L.: Real-time content transformations in a web service-based delivery architecture for geographic information. Dissertation am Finnischen Geodätischen Institut, 2007.
- [Mü08] Müller, M.: INSPIRE – Bedeutung für Dienste und Portale. In: Runder Tisch GIS e.V. (Hrsg.): Tagungsband zum 13. Münchner Fortbildungsseminar Geoinformationssysteme, München, 2008.
- [OC08] ORCHESTRA Consortium (Hrsg.): Implementation Specification of the Translating Feature Access Service, Version 0.4/1.3, 2008.
- [OGC09] Open Geospatial Consortium: <http://www.opengeospatial.org/>
- [OMG03] Object Management Group (Hrsg.): MDA Guide Version 1.0.1. OMG specification omg/2003-06-01, 2003.
- [OMG05] Object Management Group (Hrsg.): MOF 2.0 Query/Views/Transformations Specification. OMG specification ptc/05-11-01, 2005.
- [OMG07] Object Management Group (Hrsg.): UML Unified Modeling Language: Superstructure, version 2.1.1. OMG specification formal/2007-02-05, 2007.
- [SGM08] Staub, P.; Gnägi, H.R.; Morf, A.: Semantic Interoperability through the Definition of Conceptual Model Transformations. Transactions in GIS 12(2): 193-207, 2008.
- [St07] Staub, P.: A Model-Driven Web Feature Service for Enhanced Semantic Interoperability. OSGeo Journal 1(3):38-43, 2007.
- [St09] Staub, P.: Über das Potenzial und die Grenzen der semantischen Interoperabilität von

Geodaten. Dissertation, Institut für Geodäsie und Photogrammetrie, ETH Zürich, 2009.

- [VP05] Vretanos, Panagiotis A.: Web Feature Service Implementation Specification, Version 1.1.0, Open Geospatial Consortium, 2005.