

ÖREB Kataster

Schwergewichtsprojekt NW/OW „Beziehungsmodellierungssprache für verteilte Daten“

Bericht

Verfasser	Fabian Hensel, Geocloud AG
Version	1.0
Datum	05.04.2013

Einleitung

Dieses Dokument gliedert sich in zwei Teile: Im ersten Teil wird eine allgemeine Übersicht über das Schwergewichtsprojekt aus Fach- bzw. Anwendungssicht gegeben, im zweiten Teil wird die technische Implementation anhand der Referenzimplementation genauer erläutert.

Übersicht

Motivation / Problemstellung

Das Schwergewichtsprojekt der ÖREBK-Pilotkantone Nidwalden und Obwalden befasste sich mit der Modellierung der Beziehung zwischen Geodaten und Rechtsdaten. Neben den Geodaten, die den Raumbezug vorgeben und den Rechtsdaten, welche die öffentlich-rechtlichen Beschränkungen bezeichnen, muss eine dritte Art von Daten die Zuordnung zwischen diesen Daten regeln. Was bei erster Betrachtung einfach erscheint, stellt für die automatische, computergestützte Verarbeitung, etwa beim Erstellen eines ÖREBK-Auszugs, eine grosse Herausforderung dar.

Abstrakt gesehen handelt es sich um die Verknüpfung von Daten aus zwei verschiedenen Informationssystemen. Die Daten liegen also nicht in einer einfachen Datenbank vor, in der entsprechende Werkzeuge, wie etwa Views, verwendet werden können. Es bedarf deshalb eines Werkzeugs, mit welchem diese Beziehung modelliert werden kann. In der Terminologie der ÖREBK-Rahmenmodells¹ handelt es sich hierbei um die ÖREB-Definitionen bzw. den sogenannten TechOEREBCDefs, welche von den für die jeweiligen zuständigen Stellen der Geobasisdaten zur Erstellung der Transferstruktur verwendet werden.

¹

<http://www.cadastre.ch/internet/oerebk/de/home/topics/model.parsysrelated1.46997.downloadList.48207.DownloadFile.tmp/rahmenmodellvfeb2011de.pdf>

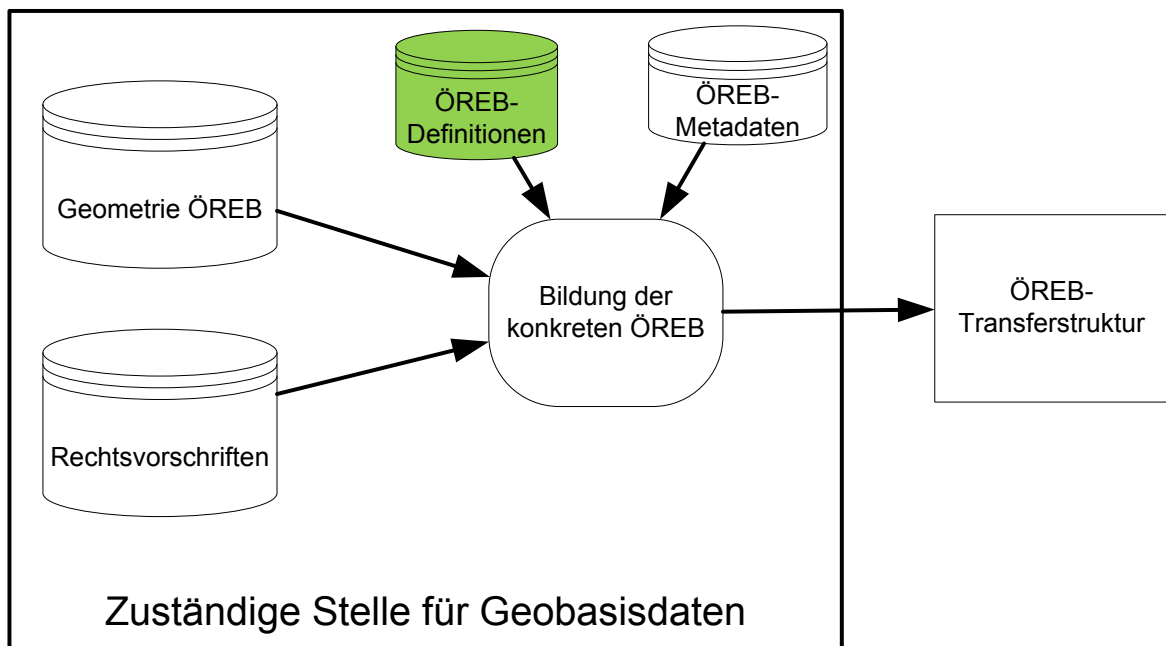


Abbildung 1: Rolle der ÖREB-Definitionen zur Bildung der konkreten ÖREB (Rahmenmodell)

Um möglichst alle Anwendungsszenarien abbilden zu können, sollte dieses Werkzeug soweit möglich abstrakt implementiert werden, dass es einerseits mit einer Vielzahl von potentiell verwendeten Informationssystemen (GIS und Rechtsdatenverwaltung) kompatibel ist, umfassende und flexible Beziehungsarten unterstützt und sich einfach in die verschiedenen ÖREBK-Softwarelösungen der Kantone einbetten lässt. Da die ÖREBK Softwarelösung der Pilotkantone Nid- und Obwalden mit der Zielsetzung einer möglichst aktuellen Auskunftserteilung, serviceorientiert aufgebaut ist, sollen die ÖREB-Definitionen neben dem durch das ÖREBK-Rahmenmodell vorgegeben Prozess der Erstellung der ÖREB-Transferstruktur durch die zuständige Stelle für Geobasisdaten auch in Echtzeit, also beim Erstellen des Katasterauszugs, prozessiert werden können.

Vorgehensweise

Aufgrund der bereits bekannten Softwarelösungen für die Geodatenverwaltung in den Kantonen Ob- und Nidwalden und der darin verwendeten Datenmodelle sowie durch die Analyse der existierenden Softwarelösungen der online verfügbaren Rechtsdatenkataloge wurden erste technische Anforderungen an die Beziehungsmodellierung gestellt. Durch die Überprüfung der ÖREB relevanten Themen gemäss Anhang der GeoIV² wurden die fachlichen Anforderungen definiert. Neben den sogenannten generell-konkreten Rechtsnormen, die mittels eines Plans

² http://www.admin.ch/ch/d/sr/510_620/app1.html

explizit die räumlichen Perimeter bezeichnen, wo diese anzuwenden sind, gibt es generell-abstrakte Rechtsnormen, welche ohne Plan definiert sind. Streng nach dem Rahmenmodell sind die generell-abstrakten Rechtsnormen zumindest in einer ersten Phase nicht Teil des ÖREB-Katasters, könnten jedoch zu einem späteren Zeitpunkt aufgenommen werden.

Bei genauerer Betrachtung von einigen Rechtsnormen, zeigte sich, dass es Rechtsnormen gibt, die zwar als Teil des ÖREB-Katasters definiert wurden, jedoch generell-abstrakter Natur sind. Als Beispiel dient die Rechtsvorschrift „Waldabstandslinien“, welche den Mindestabstand von Bauwerken zur Waldgrenze regelt. Obwohl mehrheitlich mittels eines Perimeters erfasst und damit generell-konkreter Natur, sind in den Kantonen Ob- und Nidwalden nicht alle Waldabstände explizit definiert.

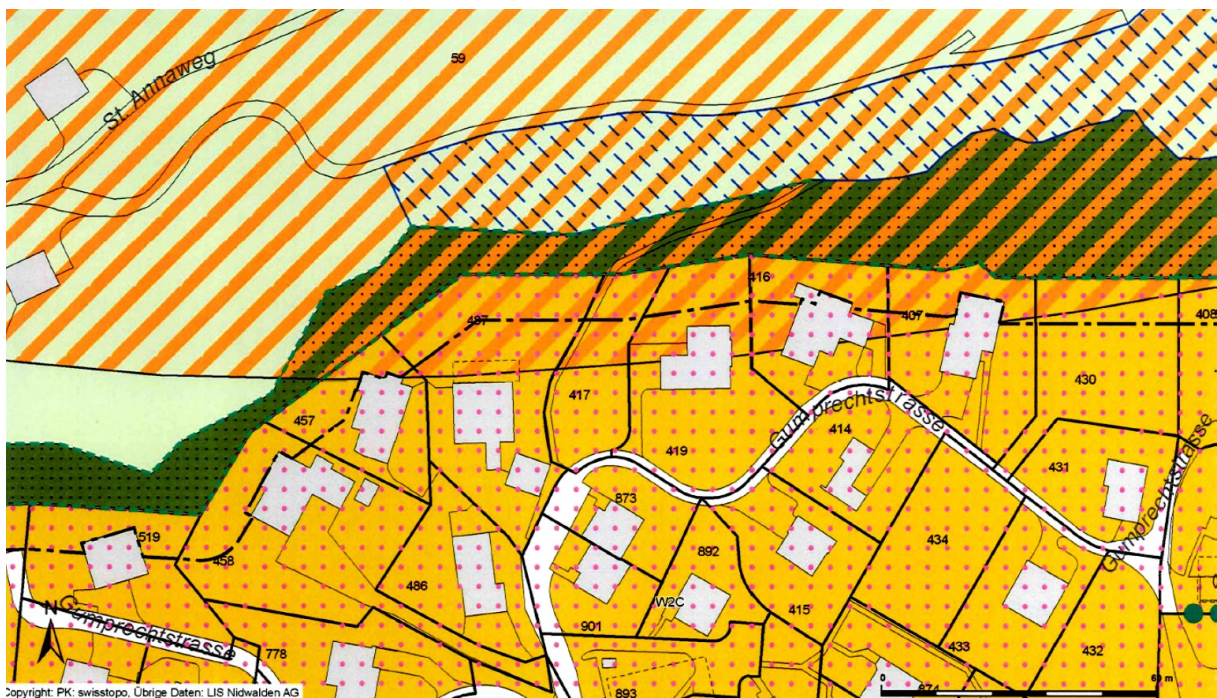


Abbildung 2: Explizit definierte Waldabstandslinie

Trotzdem gilt auch in diesem Fall für Bauwerke ein Mindestabstand von 15m zur Waldgrenze. Der Perimeter ist also nur implizit durch den Wald an sich definiert.

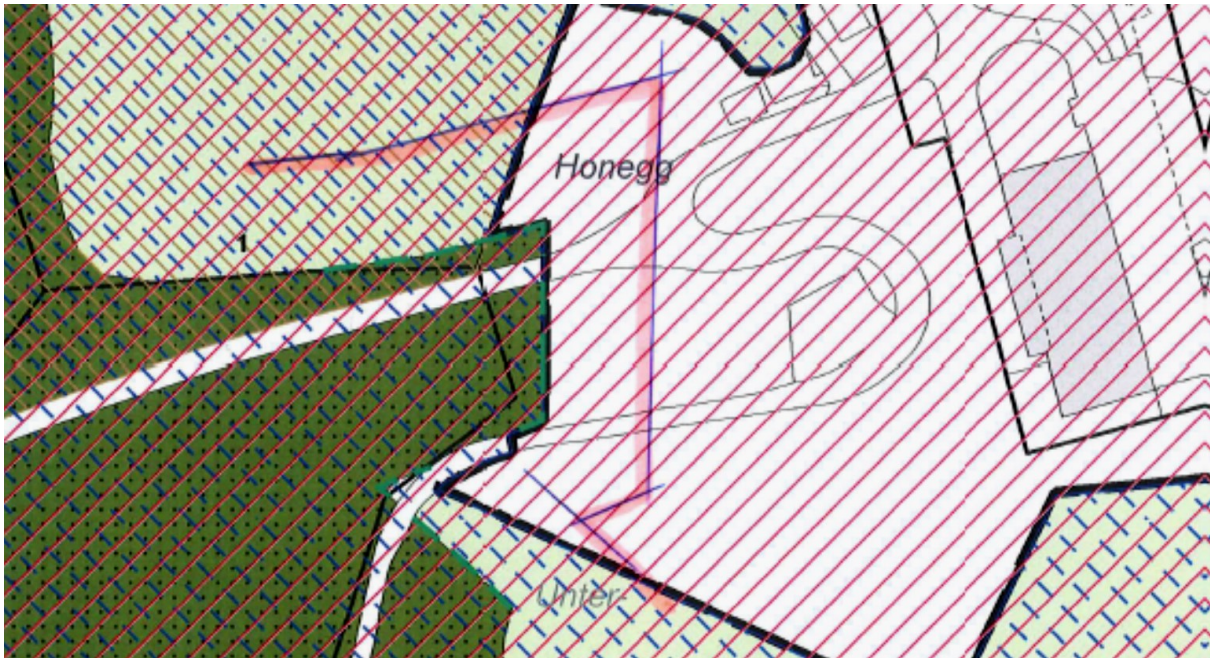


Abbildung 3: Von Hand eingezeichneter Waldabstand, welcher ohne Waldabstandslinie gilt

Katasterauszug

Aus technischer Sicht wird bei der Auswertung einer durch den Benutzer ausgewählten Liegenschaft ein Schnitt durch sämtliche ÖREB-relevanten Themenebenen die entsprechend anwendbaren Beschränkungen eruiert. Dabei gilt bei generell-konkreten Normen dass wenn sich die Geometrie der Liegenschaft mit der Geometrie des entsprechenden Perimeters schneidet (Intersect), die entsprechende Rechtsnorm angewendet wird. Die Auswertung von generell-abstrakten Normen ist nicht so einfach möglich, da eine Vielzahl von Faktoren über die Anwendbarkeit der Rechtsnorm entscheiden können. Im Beispielfall des Waldabstands ist etwa zu prüfen, ob sich in einer Distanz 15m oder weniger ein Wald befindet. Es wird also erst einmal die Geometrie der Liegenschaft um 15m gepuffert und anschliessend wiederum auf Schnitte mit Geometrien der Wälder geprüft. Diese in natürlicher Sprache einfach zu formulierende „Regel“ muss für die automatische Verarbeitbarkeit jedoch in eine technische Sprache übersetzt werden.

Prüfung Lösungsansätze

Die Umsetzung dieser Sprache sollte mit möglichst geringem Aufwand vollzogen werden und wenn möglich auf bestehende und etablierte Lösungen zurückgegriffen werden sollte, wobei neben eigentlichen „Sprachen“ auch andere Modellierungsmöglichkeiten geprüft wurden. Es zeigte sich jedoch, dass eine Sprache in Form von Zeichenketten (Strings) sehr flexibel gehandhabt und verarbeitet werden können. Geprüft wurde

unter anderem die Verwendung von GEORIF³, einer Erweiterung des Rule Interchange Formats⁴, welches für diesen Anwendungszweck vom W3C empfohlen wird. Das Format ist jedoch noch relativ neu und findet kaum praktische Anwendung. Nach Prüfung zahlreicher Alternativen wurde schliesslich JavaScript als Grundbaustein gewählt. Die Scriptsprache eignet sich besonders gut als Basis, weil sie nicht erst kompiliert werden muss, für eine Vielzahl von Anwendungen eine immer grössere Rolle spielt, leicht lesbar ist und es eine Vielzahl von Interpreten gibt, welche diese Sprache verarbeiten können.

<pre>C161e270580344121b263f4d7a52367ceN0["Art"] == "BAUZONE.W2" && isWithin(C161e270580344121b263f4d7a52367ceN0, C4ca34e6447d34aadbcbf1dcbe78db5924N0,0)</pre>	Skalarer Typfilter AND Verknüpfung Geometrischer Filter
--	---

Abbildung 4: Generell-konkret definierte Wohnzone W2

Prinzipiell werden die Regeln als boolesche Ausdrücke in JavaScript formuliert, also Aussagen, die entweder wahr (true) oder falsch (false) sein können. Dies entspricht direkt der dem Zutreffen bzw. Nicht-Zutreffen einer spezifischen Beschränkung für eine Liegenschaft. Ein Ausdruck kann jeweils aus mehreren Teilausdrücken bestehen, die normalerweise mittels des UND-Operators miteinander verknüpft sind, womit alle Teilausdrücke wahr sein müssen damit der gesamte Ausdruck wahr ist. In speziellen Fällen wäre es ebenfalls möglich Teilausdrücke mittels ODER-Operatoren zu verknüpfen womit nur einer dieser wahr sein muss, damit der Gesamtausdruck wahr ist. Teilausdrücke können grundsätzlich skalaren und geometrischen Typs sein. Für die geometrischen Vergleiche wird auf die GDAL/OGR Bibliothek⁵ zurückgegriffen, die als frei verfügbares Open Source Projekt, sowohl geometrische Manipulationen (z.B. Puffer, Konvexe Hülle, etc.) wie auch geometrische Prüfungen (z.B. Intersect) verarbeiten kann und eine Vielzahl von Geodatenformaten (u.A. INTERLIS) unterstützt. Damit kann beispielsweise festgelegt werden, dass sich eine Liegenschaft mit einer Nutzungszone schneiden muss, damit der (Teil-)ausdruck wahr ist. Andererseits beziehen sich die skalaren Vergleiche auf Attribute von Objekten (im GIS Features) mit in der Regel vorgehaltenen Sollwerten. Somit kann der Ausdruck beispielsweise konkretisiert werden, indem eine Rechtsnorm nur für Liegenschaften in der Nutzungszone „W2“ gilt. In diesem Fall werden die beiden Teilausdrücke UND verknüpft.

³ <http://www.wichmann-verlag.de/gis-fachzeitschriften/artikelarchiv/2011/gis-science-ausgabe-032011/zum-austausch-raeumlicher-regeln-auf-dem-weg-zu-einem-geobasierten-regelaustauschformat-georif.html>

⁴ <http://www.w3.org/standards/techs/rif>

⁵ <http://www.gdal.org/ogr/>

<pre> C161e270580344121b263f4d7a52367ceN0["Art"] == "NICHTBAUZONE.Wald" && isWithin(C161e270580344121b263f4d7a52367ceN0, C4ca34e6447d34aadbcbf1dcbe78db5924N0,15) </pre>	Skalarer Typfilter AND Verknüpfung Geometrischer Filter
--	---

Abbildung 5: Generell-abstrakt definierter Waldabstand in Regelausdruckssprache

Die JavaScript Ausdrücke können prinzipiell von einem beliebigen JavaScript Interpreter, wie diese beispielsweise in jedem Internet Browser vorkommen, evaluiert werden. Für die Referenzimplementation, die für die auf Microsoft .NET aufbauende ÖREBK-Applikation der Kantone Ob- und Nidwalden entwickelt wurde, wird jedoch JINT⁶ als Interpreter verwendet. JINT ist ein JavaScript Interpreter der transparent aus .NET-Anwendungen aufgerufen werden kann und umgekehrt auch transparente .NET-Methodenaufrufe aus JavaScript unterstützt. Dies ist Voraussetzung für die nahtlose Integration von GDAL/OGR, da die Bibliothek keine native JavaScript Schnittstelle anbietet. Stattdessen werden die geometrischen Operationen über den Umweg von .NET aufgerufen. Grundsätzlich können alle Funktionen der GDAL/OGR Bibliothek genutzt werden. In der Referenzimplementation ist der häufigste Fall, die Abstandsprüfung, mittels der Funktion „IsWithin“ abgebildet, mit welcher das Zutreffen einer Rechtsnorm anhand des Abstand zwischen zwei Geometrien geprüft werden kann. Im konkreten Beispiel für etwa für die Prüfung ob sich ein Waldgebiet in 15m Abstand einer Liegenschaft befindet. Generell-konkrete Rechtsnormen lassen sich ebenfalls mit Hilfe des „IsWithin“ Operators verknüpfen indem der Distanzparameter ganz einfach auf 0m gesetzt wird. Auf die Attribute der Objekte wird bei skalaren Teilausdrücken mittels der Indexer (z.B. ["Art"]) zugegriffen. Sollten diese beiden Arten von Ausdruckstypen in Zukunft nicht genügen, weil etwa das Zutreffen einer Beschränkung von anderen Faktoren abhängig ist, so lassen sich beliebig komplexe Ausdrücke durch die Anmeldung von Prozeduren entweder in .NET oder JavaScript, formulieren. Die Beziehungsmodellierungssprache ist also unbeschränkt erweiterbar.

Um die verschiedenen Eingabemengentypen , in Falle der GIS Applikation, den Layers, entsprechend in den Regeln bezeichnen zu können, wird jedem Eingabemengentyp ein eindeutiger Identifikator in Form einer sogenannte GUID zugeordnet (z.B. C161e270580344121b263f4d7a52367ce für die Nutzungsplanung). Durch weitere Abstraktion ist es möglich, die Nutzungspläne der verschiedenen Gemeinden unter einem Metaidentifikation zusammenzufassen, womit die in der Regel enthaltenen Identifikatoren

⁶ <http://jint.codeplex.com/>

bei der Auswertung erst in die entsprechenden lokal gültigen Identifikationen übersetzt werden. Dem eindeutigen Identifikator folgend ist ein Aufzählungswert (N0 bis N9), welcher die Verwendung unterschiedlicher Objekte des gleichen Eingabemengentyps in einer Regel zulässt. Dies ist nur bei äusserst komplexen Regeln erforderlich, weshalb meist der Wert N0 verwendet wird.

Die Erfassung und Bearbeitung von Regeln wird durch ein Editierwerkzeug unterstützt, welches die verfügbaren Eingabetypmengen und deren Attribute in Auswahllisten darstellt. Einfache Regeln können so auch durch Personen, die nicht mit der Ausdruckssprache vertraut sind, erfasst werden.

Verarbeitung

Die Verarbeitung der Regeln wird in der sogenannten „Rule Engine“ durchgeführt, welche einerseits auf Abruf, etwa wenn der Benutzer eine Auswertung startet, anderseits auch im Stapelverarbeitungsmodus betrieben werden kann (Bildung der konkreten ÖREB). Als Eingabewerte erwartet die „Rule Engine“ die zu verarbeitenden Objekte (Features) und eine Liste von zu prüfenden Regelausdrücken, die im Regelfall von einer Datenbank bezogen werden. Beim Betrieb auf Abruf wird der „Rule Engine“ die durch den Benutzer ausgewählte Liegenschaft übergeben, sowie Schnittobjekte der relevanten ÖREB-Themen im Umkreis von 100m. Dieser Distanzwert ist arbiträr gewählt und muss gegebenenfalls später noch angepasst werden. Die „Rule Engine“ verarbeitet jede einzelne Regel nach einander, wobei dieser Prozess auch parallelisiert durchgeführt werden kann, um die Auswertungsgeschwindigkeit zu erhöhen. Für jeden Regelausdruck werden zuerst anhand der eindeutigen Identifikatoren die benötigten Eingabemengentypen bestimmt (z.B. Liegenschaft und Nutzungszonen). Anschliessend werden alle möglichen Kombinationen der möglichen Eingabemengen anhand des kartesischen Produkts bestimmt. Als zentraler Schritt wird nun mittels JINT die Regel evaluiert. Falls der Ausdruck „wahr“ ist, wird die mit der Regel verbundene Rechtsnorm der Ausgabemenge hinzugefügt. Wenn alle Regeln abgearbeitet sind, haben wir eine Liste von für die entsprechende Eingabemenge (Liegenschaft) anwendbaren Rechtsnormen.

Fazit

Mit Hilfe der auf Standardtechnologien entwickelten Beziehungsmodellierungssprache ist es möglich systemübergreifend die Zusammenhänge zwischen Geo- und Rechtsdaten zu erfassen. Die festgelegte Regelausdruckssprache ermöglicht die Modellierung anhand skalarer und räumlicher Aspekte. Einfache Regeln für generell-konkrete

Rechtsnormen bis zu beliebig komplexen Regelausdrücken durch die Verkettung von Teilausdrücken sind möglich. Durch die Verwendung von Open Source Technologien entstehen keine weiteren Kosten. Die Erfassung von Beziehungen wird mittels eines Editierwerkzeugs unterstützt und lässt sich prinzipiell in jede bestehende GIS- und Rechtsdatenverwaltungssoftwareumfeld einbetten.

Technische Dokumentation Referenzimplementation

Sprache

Die Regelausdruckssprache entspricht booleschen JavaScript Ausdrücken und ist daher über die ECMAScript⁷ Ausdrucksspezifikation definiert. Die Ausdrücke werden mittels der JavaScript Engine *Jint*⁹ evaluiert. Grundsätzlich können also alle gültigen JavaScript Ausdrücke als Regel verwendet werden. So kann z.B. eine Regel, die in jedem Fall wahr ist einfach durch den Ausdruck `"true"` angegeben werden. Komplexe Regelausdrücke können durch die Verwendung von AND (`&&`) und OR (`||`) Operatoren verfasst werden. Bei Verwendung des „AND“ Operators wird die Regel von links nach rechts nur so lange evaluiert, wie alle Teilausdrücke wahr sind. Sobald ein Teilausdruck falsch ist, wird der gesamte Ausdruck falsch. Um eine möglichst hohe Verarbeitungsgeschwindigkeit zu erzielen, ist es deshalb ratsam, einfach zu prüfende Regelemente an den Anfang des Regelausdrucks zu platzieren.

Da einfache Ausdrücke ohne Miteinbezug externer Daten im Nutzen sehr beschränkt sind, konzentriert sich die Implementation auf deren Einbindung. Um verschiedene Klassen von Daten zu unterscheiden, werden GUIDs als Klassenidentifikator verwendet. Beispiele von Datenklassen sind etwa Liegenschaften, Nutzungszonen und Gesetzesartikel. Alle Instanzen einer Datenklasse besitzen dieselben Attribute, können jedoch unterschiedliche Attributwerte enthalten. Natürlich ist die `ClassId` aller Instanzen einer Datenklasse identisch. Weiter erlaubt das Attribut `geometry` die Verwaltung von Geometrien. Generisch als Datentyp `string` angegeben, kann die Geometrie in einem beliebigen von GDAL/OGR unterstützten Format angegeben werden (z.B. GeoJSON, WKT, GML). Das Attribut `ObjectIdAttributeName` gibt den Namen des Attributs an, welches den eindeutigen Identifikator der Instanz enthält. Dieser lässt sich einfach mittels der Accessors der `ObjectId` Attributs lesen und schreiben. Die Gesamte Klasse `Feature`

⁷ <http://www.ecma-international.org/publications/standards/Ecma-262.htm>, Kapitel 11

⁹ <http://jint.codeplex.com/>

kann prinzipiell als generisches, abstrahiertes Objektmodell betrachtet werden. Die umliegenden Systeme (GIS / Rechtsdatenverwaltung) müssen ihre erst Objekte in dieses generische Format übersetzen, um diese dem `Oereb.Evaluator` übergeben zu können.

Damit in einer Regel mehr als eine Instanz einer Datenklasse in Sinne eines Parameters referenziert werden kann (z.B. zwei verschiedene Zonentypen, beides Instanzen der Datenklasse Nutzungszonen), werden die GUIDs mittels eines Zählers ergänzt, welcher die Werte 0 bis 9 enthalten kann. Der entsprechende reguläre Ausdruck für die Spezifikation von Datenklassen in Regelausdrücken ist wie folgt definiert:

```
C[0-9a-fA-F]{32}N[0-9]
```

Ein gültiges Beispiel ist

```
C3F2504E04F8911D39A0C0305E82C3301N0
```

Die Zeichen **C** und **N** sind statisch und dienen ausschliesslich der besseren Lesbarkeit.

Während der Regelevaluation wird für jeden dieser Datenklassenplatzhalter ein entsprechender Parameterwert eingefügt.

Implementierte Filter

In der Referenzimplementation enthalten sind zwei verschiedene Arten von Filtern, einerseits klassische Vergleichsoperatoren auf nicht-geometrische Attribute, andererseits ein Abstandstest zweier Geometrien.

Nicht-geometrischer Vergleich

Nicht-geometrische Vergleiche können einfach mittels des Indexers des Parameters formuliert werden und mit einem statisch definierten Wert oder einem Attributwert eines anderen Parameters verglichen werden.

Beispiele:

```
C161e270580344121b263f4d7a52367ceN0["Art"] ==  
"NICHTBAUZONE.Wald"
```

```
C7629098c373240539815a3308fad394eN0["Baujahr"] ==  
C38ae98a39f074d6591f886d7820c4b26N0["Baujahr"]
```

Geometrischer Abstandstest

Der geometrische Abstandstest erfolgt durch Aufruf der Methode `isWithin([Geometrie1], [Geometrie2], [AbstandInMeter])`. Obwohl die Methode in .NET in der Klasse `Evaluator` implementiert ist und von dort GDAL/OGR referenziert, um die eigentliche Berechnung mittels

einer Puffer und anschließenden Intersection-Operation durchzuführen, kann diese vom JavaScript Code aus aufgerufen werden. Diese nahtlose Konvergenz zwischen .NET und JavaScript wird erreicht, indem die `isWithin` Methode bei der Initialisierung von *Jint* mittels der Methode `SetFunction` angemeldet wird. Mittels eines Delegate werden die Ein- und Ausgabeparameter der JavaScript Methode definiert. Mehr Informationen sind in der Jint Dokumentation zu finden¹⁰.

Beispiel:

```
isWithin(C161e270580344121b263f4d7a52367ceN0,  
C4ca34e6447d34aadbcf1dcbe78db5924N0,15)
```

Ausbaubarkeit

Alleine mit dem nicht-geometrischen Vergleich können alle Typ und Id basierten Filter für generell-konkrete Rechtsnormen im Sinne des ÖREB-Rahmenmodells abgebildet werden. Der geometrische Abstandstest erlaubt Abhängigkeitskriterien für generell-abstrakte Rechtsnormen zu formulieren. Mit diesen beiden Filtern kann der allergrösste Teil der zu erwartenden Regeln formuliert werden. Sollte sich jedoch herausstellen, dass die Funktionalität ungenügend ist, so kann diese jederzeit erweitert werden.

Programmbibliothek

Aufbau

Das Projekt `Oereb.Evaluator` besteht zum einen aus der Hauptklasse `Evaluator` und der Hilfsklasse `Feature` sowie einer statischen Klasse für `Extensions des IEnumerable Typs`. Prinzipiell wird die `Evaluator` Klasse vom Hauptprogramm instanziiert und die `Evaluation` Methode für jeden Regelausdruck in der Regel innerhalb einer Schleife aufgerufen.

Eingabe

Methodensignatur:

```
public List<List<Feature>> Evaluation(List<Feature> features,  
string expression, double ruleOffset)
```

Die Regelausdrücke werden als `String` übergeben (`expression`) und können daher in verschiedenen Formen gespeichert werden, etwa in einer Datenbank oder in einer XML-Datei. Neben dem Regelausdruck wird eine Liste der zu verarbeitenden Features mit deren Geometrien

¹⁰

<http://jint.codeplex.com/wikipage?title=Passing%20external%20functions%20to%20scripts&referringTitle=Documentation>

übergeben. Diese beinhaltet unter anderem die Liegenschaft für welche die Auswertung gewünscht wird, andererseits aber auch alle Features der Layers, welche die Liegenschaft schneiden (intersect). Damit generell-abstrakte Rechtsnormen ebenfalls verarbeitet werden können, muss die Liste nicht nur Features enthalten, welche die Liegenschaft unmittelbar schneiden, sondern auch jene die sich in einem bestimmten Umkreis der Liegenschaft befinden. Die Erzeugung dieser Featureliste ist Teil der GIS-Infrastruktur und beinhaltet im Wesentlichen eine Pufferoperation der Liegenschaftsgeometrie mit der Verschneidung der Schnittebenen. Es ist möglich die `Evaluation` Methode in mehreren Threads gleichzeitig aufzurufen, um durch Parallelisierung eine höhere Verarbeitungsgeschwindigkeit zu erzielen. Der `ruleOffset` Parameter lautet normalerweise 0. Durch die Angabe von anderen, positiven oder negativen, Werten ist es möglich die Abstandprüfung zweier Featuregeometrien in der `IsWithin` Methode entsprechend zu beeinflussen.

Verarbeitung

In einem ersten Schritt werden aus dem übergebenen Regelausdruck (`expression`) die darin enthaltenen Parameter extrahiert. Dies geschieht mittels der Methode `GetFeatureParametersFromRule` durch Verwendung eines regulären Ausdrucks. Die Methode retourniert die Liste aller in der Regel enthaltenen Parameter, die für die Prozessierung der Regel befüllt werden müssen. In einem nächsten Schritt werden mittels der Methode `GetPossibleFeatureValuesForParameters` alle möglichen Kombinationen aus Features bestimmt, welche für die jeweiligen Parameter in Frage kommen. Ob ein Feature für einen Parameter in Frage kommt wird anhand der `parameterClassId` bestimmt, die im Abschnitt "Sprache" erläutert wird. Als Ausnahmebehandlung wird die Prozessierung abgebrochen, wenn für mindestens einen der Parameter nicht mindestens ein passender Wert gefunden werden konnte. In diesem Fall kann der Regelausdruck nie wahr sein, weshalb sich die restliche Verarbeitung erübrigt. In allen anderen Fällen wird nun das kartesische Produkt (Kreuzprodukt) aller möglichen Werte die jeweiligen Parameter des Ausdrucks bestimmt, womit in der Folge alle Parameterwertkombinationen in einer Schleife geprüft werden.

Jint wird ggf. nach Singleton Pattern vor der Schleife erst initialisiert. Innerhalb der Schleife wird in einer zweiten Schleife jeder Parameter mit dem entsprechenden Wert mittels `SetParameter` bei *Jint* angemeldet. Sind alle Parameterwerte gesetzt, wird der Ausdruck mittels `Run` evaluiert. Ist die Parameterwertekombination für den Regelausdruck wahr, so wird diese der Ausgabeliste hinzugefügt. Wenn der

Regelausdruck falsch ist oder wenn bei der Verarbeitung ein Fehler unterläuft, werden keine weiteren Aktionen durchgeführt.

Ausgabe

Die Ausgabe der `Evaluate` Methode erfolgt als eine Liste von Listen von Parameterwertkombinationen in Form von `Feature` Objekten, für welche der eingegebene Regelausdruck für die eingegebene Eingabemenge wahr ist.