



DLR

**Deutsches Zentrum
für Luft- und Raumfahrt e.V.**
in der Helmholtz-Gemeinschaft



Integration energiemeteorologischer Daten in Geographische Informationssysteme

Arbeitspaket 2.3

Sönke Brummerloh
OFFIS e.V.
Escherweg 2
26121 Oldenburg

Telefon: +49 441 9722 – 189
Telefax: +49 441 9722 – 102
E-Mail: soenke.brummerloh@offis.de

1 Einleitung

Im Rahmen der energiemeteorologischen Arbeiten bei DLR-TT hat sich in den letzten Jahren eine Vielzahl heterogener geographischer Daten angesammelt. Dazu gehören klassische Datensätze zur Landnutzung und -beschaffenheit, energiespezifische Datensätze über die Strom- und Gasinfrastruktur sowie zahlreiche Datensätze zu verfügbaren erneuerbaren Ressourcen. Um diese Datensätze weiterhin effizient nutzen zu können, wird eine angemessene Dokumentation und Indizierung benötigt. In AP 2.3 (Integration energiemeteorologischer Daten in Geographische Informationssysteme) wurde daher ein Datenmanagementkonzept und ein entsprechendes Datenmanagementsystem erstellt, um relevante Datensätze effizient verwalten und rasch auffinden zu können.

Das zugrunde liegende Konzept bei der umgesetzten Lösung ist, dass beliebige Dateien zu Datensätzen zusammengefasst und diese mit Metadaten versehen werden können. Metadaten über die zu einem Datensatz gehörenden Dateien, werden in einer XML-Datei gespeichert. Wenn Datensätze auf Dateien aus Geoinformationssystemen (GIS-Dateien) basieren, werden bereits vorliegende Metadaten übernommen. Da die Daten oft in Microsoft Windows basierten GIS-Anwendungen verarbeitet werden, wurde eine Lösung umgesetzt, die es ermöglicht auch weiterhin über eine herkömmliche Verzeichnisstruktur auf einem Netzlaufwerk auf die Dateien zu zugreifen.

Die einzelnen durchgeführten Teilschritte in AP 2.3 waren:

1. Analyse der bisher genutzten GIS-Dateiformate und ihrer Metadaten. Im Einzelnen waren dies die Formate ArcGIS, IDRISI und SOLEMI.
2. Definition eines XML-basierten Datenformats zur Beschreibung der einzelnen Datensätze.
3. Für das XML-basierte Metadatenformat wurden Lese- und Schreiboperationen in C und Perl bereitgestellt. Bei diesen Bibliotheken ist der Zugriff sowohl auf lokale Dateien, als auch über Netzwerk-Protokolle auf entfernte Ressourcen möglich.
4. Auf Grundlage des XML-basierten Metadatenformats wurde eine Anwendung umgesetzt, die einen Suchindex und einen Katalog für die Datensätze bereitstellt. Die Suchkriterien stützen sich dabei auf die in den XML-Dateien gespeicherten Metainformationen. Über diese Anwendung ist es möglich Datensätze zu erstellen und zu durchsuchen. Soweit möglich, werden Metadaten aus der GIS-Datei, auf der ein Datensatz basiert, extrahiert und in den Datensatz automatisch übernommen.

2 Metadatenformat

Das GIS-Metadatenformat ist angelehnt an den Standard ISO 19115, der ebenfalls ein Metadatenformat für GIS-Daten beschreibt. Aus Kompatibilitätsgründen ist eine Verwendung oder zumindest Anlehnung an den ISO 19115 Standard sinnvoll. Allerdings ist das in der ISO 19115 spezifizierte Schema mit ca. 300 Elementen sehr umfangreich, enthält aber trotzdem nicht die

Möglichkeit alle Metadaten der zu berücksichtigenden GIS-Formate (IDRISI, SOLEMI und ArcGIS) zu speichern. Daher wurde versucht das GIS-Metadatenchema dem Standard zwar möglichst weit anzunähern, aber ohne dabei das neue Schema unnötig zu verkomplizieren. Aus diesem Grund wurde der Standard an vielen Stellen stark vereinfacht. Zusätzlich wurde das Element `dataInfo` eingeführt, um Daten aus den verwendeten Formaten unkompliziert übernehmen zu können.

Das Metadatenformat ist ein XML-basiertes Format, dass mit XML-Schema definiert wurde.

Auf der obersten Ebene finden sich folgende Elemente:

Element	Typ	Beschreibung
language	Zeichenkette	Dokumentationssprache in der die Texte verfasst sind. Wird gesetzt falls die Sprache nicht über <code>xml:lang</code> gesetzt wurde. (Format: ISO 639-1)
characterSet	Zeichenkette	Zeichenkodierung der Metadaten (z.B. UTF-8, UTF-16, ISO-8859-1, ISO-8859-2). Muss verwendet werden falls es nicht im Kopf der XML-Datei definiert wurde oder das Format ISO/IEC 10646-1 nicht verwendet wird.
contacts	Contacts	Kontaktinformationen zu den Verantwortlichen der Metadatendatei
dataStamp	Datum	Erstellungsdatum der Metadatendatei
metadataStandardName	Zeichenkette	Name des Metadatenstandards (hier: vIEM-GIS-Metadata)
metadataStandardVersion	Zeichenkette	Versionsnummer des Metadatenstandards (hier: 1.0)
IdentificationInfos	IdentificationInfos	Basisinformationen zum Datensatz, wie z.B. ein Abstract, Schlagwörtern oder verantwortliche Personen.
distributionInfo	DistributionInfo	Informationen zum Datensatzformat und Bezugsquellen, wie z.B. das Format der GIS-Datei auf der dieser Datensatz basiert und die Dateien die zu diesem Datensatz gehören.
dataQualityInfo	DataQualityInfo	Textuelle Informationen zur Qualität der Daten, die aus den GIS-Dateien importiert wurden.
dataInfo	DataInfo	Metadaten, die größtenteils aus den GIS-Dateien extrahiert wurden.
additionalFiles	AdditionalFiles	Dateien die dem Datensatz zusätzliche zugeordnet sind.

Auf eine detailliertere Beschreibung des Metadatenchemas wird an dieser Stelle verzichtet. Eine detaillierte Beschreibung ist im GIS-Metadata-Manager-Paket auf <http://sourceforge.net/projects/gis-manager> zu finden.

3 Parser-Bibliotheken

Wünschenswert war es, Bibliotheken zu entwickeln, um Dateien im Metadatenformat möglichst einfach lesen und schreiben zu können. Da beim DLR-TT vor allem die Programmiersprachen C und Perl eingesetzt werden, wurde die Parser-Bibliothek in diesen beiden Sprachen implementiert. Beide Bibliotheken sind in der Lage GIS-Metadatendateien vollständig einzulesen und in interne Speicherstrukturen zu überführen. Dabei ist es auch möglich, eine Datei über ein Netzwerkprotokoll wie HTTP oder FTP zu laden. Die internen Speicherstrukturen können modifiziert und wieder in eine GIS-Metadatendatei gespeichert werden.

Einzelne Elemente können aus der GIS-Metadatendatei über geeignete Pfadangaben im XPath-Format¹ ausgelesen werden. Auf diese Weise ist es möglich, gezielt einzelne Elemente abzuändern. Die interne Speicherstruktur kann jederzeit gegen das XML-Schema validiert werden.

Die Parser-Bibliothek in C basiert auf den frei verfügbaren Bibliotheken libxml2² und cURL³ und wird wie eine typische Linux-Bibliothek kompiliert und installiert mit:

```
./configure
```

```
make
```

```
make install
```

Nach der Installation sollte einmal der Befehl `/sbin/ldconfig` ausgeführt werden, um die neue Parser-Bibliothek verfügbar zu machen (wird normalerweise beim Starten von Linux automatisch durchgeführt). Danach kann die Parser-Bibliothek wie eine normale Bibliothek unter Linux in einem C-Programm verwendet werden.

Folgendes Beispiel demonstriert die Funktionsweise der Parser-Bibliothek:

```
#include <gisparser.h>

int main(int argc, char *argv[]) {
    // Before any function from the GISParser is used,
    // startGISParsing() has to be called!
    startGISParsing();

    // Note: The GISParser is working with the xmlDocPtr struct
    // from libxml2, so the functions from libxml2 can be used
    // on xmlDocPtr.
    xmlDocPtr doc;

    // First load a GIS-Metadatei file into a XML document
    doc = loadGISMetadatei("test.xml");

    // Now the data in the XML document can be modified
    setValue(doc, "//GIS_Metadatei/metadataStandardVersion", "1.23");
}
```

1 <http://www.w3.org/TR/xpath>

2 <http://xmlsoft.org>

3 <http://curl.haxx.se>

Integration energiemeteorologischer Daten in Geographische Informationssysteme

```
// After modifications, the XML data can be saved again
saveGISMetadata(doc, "test_out.xml");

// Some cleaning up:
xmlFreeDoc(doc);

// At the end stopGISParsing() has to be called!
stopGISParsing();

return 0;
}
```

Wie beim Parser in C werden in der Perl-Version XPath-Ausdrücke verwendet und im Hintergrund mit den gleichen Bibliotheken gearbeitet. Daher werden die beiden Perl-Bibliotheken XML::LibXML und WWW::Curl::Easy benötigt.

Die Verwendung der Parser-Bibliothek in Perl gleicht der in C, ist aber noch etwas prägnanter, wie folgendes Beispiel zeigt:

```
use strict;
use warnings;

use GISParser;

# First: load an existing GIS-Metadata file
my $doc = GISParser::loadFromFile("test.xml");

# Now the data can be modified
GISParser::setValue($doc, "//GIS_Metadata/metadataStandardVersion", "12.3");

# The xml document can be checked for validity to an xml schema
if (GISParser::validateXML($doc, "schema/gis-metadata-schema.xsd"){
    print "GIS-Metadata is valid\n"
}
else {
    print "GIS-Metadata is not valid\n";
}

# After the data was modified it can be saved into a file
GISParser::saveToFile($doc, "modified_gisdata.xml");
```

4 GIS-Metadatenmanagement

Zum Arbeiten und Verwalten der Datensätze mit ihren Metadaten wurde eine Java-basierte Desktop-Anwendung mit dem Namen GIS-Metadata-Manager implementiert. Mit dieser ist es möglich mehrere Dateien zu einem Datensatz zusammenzufassen und diesen Datensatz mit Metadaten zu versehen. Dabei werden vorhandene Metainformationen aus bereits vorliegenden Daten extrahiert und in die XML-Metadaten-Datei übernommen. Die Metadaten aller Datensätze können in der Such-Ansicht nach Schlüsselwörtern durchsucht werden. Darüber hinaus kann eine Auswahl an Datensätzen auf Basis der Metadaten in der Katalog-Ansicht Katalog-artig immer weiter eingeschränkt werden.

Beim Einsatz der Anwendung bleibt die bestehende Verzeichnisstruktur des verwendeten Datenspeichers erhalten und wird nur um relativ kleinen XML-Dateien

ergänzt. Da es hilfreich ist, Dateien auch manuell verwalten zu können, können Dateien und Datensätze mit dem GIS-Metadaten-Manager verschoben, kopiert und gelöscht werden, sowie aus der Verzeichnisstruktur heruntergeladen werden.

Externe Änderungen im Dateisystem können automatisch vom GIS-Metadaten-Manager erkannt werden. Dies ermöglicht es mehreren Nutzern, in gewissem Umfang parallel auf den selben Dateien zu arbeiten. Falls eine Datei bzw. ein Datensatz allerdings von mehreren Nutzern zur gleichen Zeit geändert wird, ist es wahrscheinlich, dass es zu einem Datenverlust kommt. Um dem entgegen zu wirken, können Verzeichnisse, Dateien und Datensätze für andere Nutzer innerhalb des GIS-Metadaten-Managers gesperrt werden.

Der GIS-Metadaten-Manager basiert auf Java⁴ 1.5, Groovy⁵ 1.5 und Qt Jambi⁶ 4.4. Auf Grund der Lizenzbedingungen von Qt Jambi steht der GIS-Metadaten-Manager unter der GNU General Public License (GPL⁷).

Die Programmiersprache Groovy⁸ wurde für Plugins verwendet mit denen Metadaten aus vorhandenen GIS-Dateien extrahiert werden. Groovy ist eine an Java angelehnte dynamische Skriptsprache. „Angelehnt“ heißt in diesem Fall, dass Java-Quellcode gültiger Groovy-Quellcode ist (aber in der Regel nicht andersherum) und alle Java-Bibliotheken nutzen kann. Wenn gewünscht, kann in Groovy auf weiterführende Konstrukte und einen gegenüber Java einfachere Syntax zurückgegriffen werden. Ein Grund aus dem Groovy zum Implementieren der Plugins im GIS-Metadaten-Manager verwendet wird, ist, dass Groovy-Dateien ausgeführt werden können, ohne dass sie vorher, wie in Java notwendig, explizit in class-Dateien kompiliert werden müssen. Da die Plugins als Quellcode-Dateien vorliegen können unkompliziert neue Plugins entwickelt und dazu bereits vorhandene Plugins als Grundlage verwendet werden.

Qt Jambi ist eine Bibliothek, die es ermöglicht rasch performante grafische Oberflächen (GUIs) zu erstellen. Entwickler werden durch die durchdachte API (Application Programming Interface) und die mitgelieferten Werkzeuge, beispielsweise zum Erstellen von GUIs⁹ und zum Übersetzen der Oberflächenbeschriftungen¹⁰, sehr gut unterstützt.

Der GIS-Metadaten-Manager wurde über mehrerer Prototypen hinweg entwickelt. Für jeden Prototypen wurde vor dessen Implementierungsphase die zu integrierende Funktionalität festgelegt und im WISENT-Wiki aufgelistet. Jeder Prototyp wurde im Wiki veröffentlicht und damit DLR-TT zur Begutachtung zur Verfügung gestellt. Durch kurze Release-Zyklen konnten schon früh in der Entwicklung Rückmeldungen gegeben werden und festgestellt werden, was nicht den Erwartungen entsprechend umgesetzt wurde. Auf diese Weise konnten auch fehlende Funktionalitäten identifiziert werden, die zu Beginn noch nicht geplant waren, aber das Arbeiten mit dem GIS-Metadaten-Manager verbesserten.

Zum Starten des GIS-Metadaten-Managers unter Microsoft Windows und Linux können die Dateien start.bat bzw. start.sh genutzt werden. Unter Microsoft

4 <http://java.sun.com>

5 <http://groovy.codehaus.org>

6 <http://troll.no/products/qt/jambi>

7 <http://www.gnu.org/copyleft/gpl.html>

8 <http://groovy.codehaus.org>

9 http://doc.trolltech.com/qtjambi-4.3.4_01/doc/html/com/trolltech/qt/qtjambi-designer.html

10 http://doc.trolltech.com/qtjambi-4.3.4_01/doc/html/com/trolltech/qt/linguist-manual.html

Windows kann der GIS-Metadaten-Manager auch über eine exe-Datei gestartet werden. Prinzipiell kann der GIS-Metadaten-Manager auch unter MacOS X mit einem wie in start.sh verwendeten Aufruf gestartet und genutzt werden. Lediglich die für MacOS X notwendigen Qt Jambi Bibliotheken müssen beim Aufruf angegeben werden.

Der GIS-Metadaten-Manager gliedert sich in vier Hauptoberflächen und mehrere Dialoge. Im Folgenden werden diese Hauptoberflächen und der Metadaten-dialog vorgestellt.

4.1 Datei-Ansicht

Die Datei-Ansicht zeigt alle Verzeichnisse und Dateien bis auf die XML-Metadaten-dateien an.

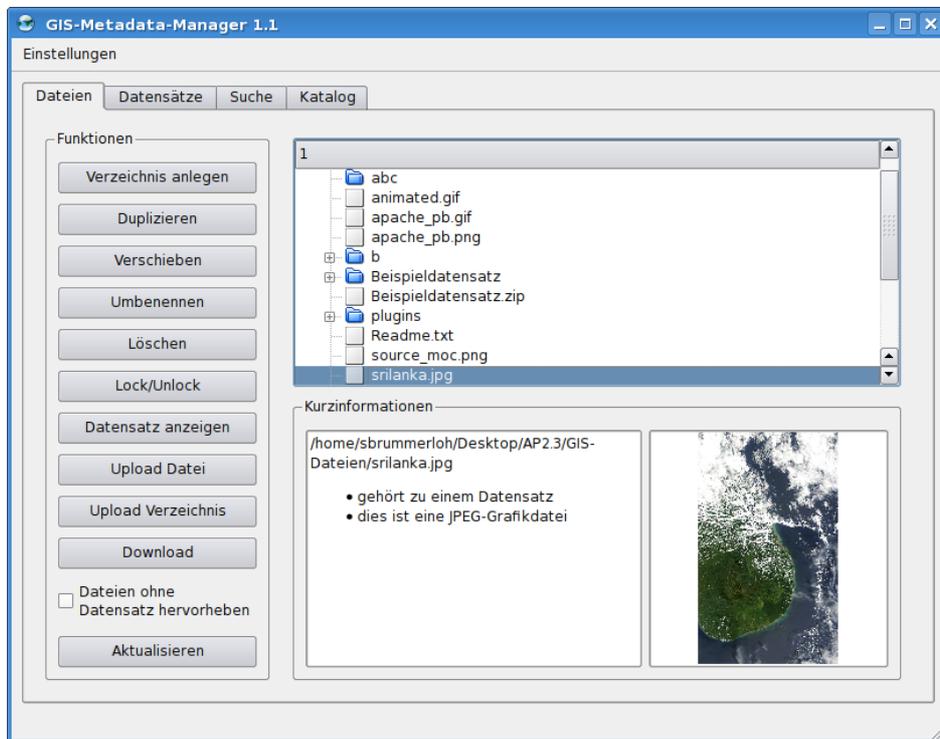


Abbildung 1: Die Datei-Ansicht

Rechts oben wird ein Dateibaum angezeigt. Rechts unten werden Informationen zu dem im Dateibaum selektierten Verzeichnis bzw. der selektierten Datei angezeigt. Falls es sich bei der selektierten Datei um eine Grafikdatei handelt, wird rechts unten eine Vorschau angezeigt. Ein Doppelklick auf die Vorschau-Datei öffnet einen Dialog in dem die Grafik in Originalgröße angezeigt wird. Auf der linken Seite sind verfügbare Befehle aufgelistet. Dort befindet sich auch eine Option mit der alle Dateien, die nicht einem Datensatz zugeordnet sind, hervorgehoben werden.

4.2 Datensatz-Ansicht

In der Datensatz-Ansicht werden nur Verzeichnisse und XML-Metadatendateien angezeigt.

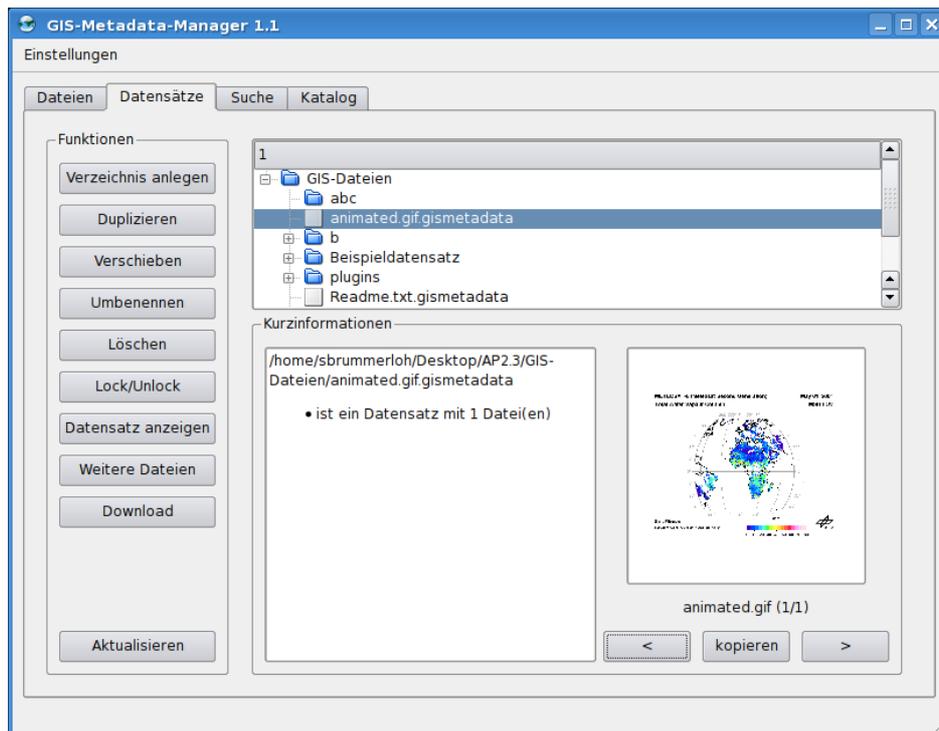


Abbildung 2: Die Datensatz-Ansicht

Rechts oben wird ein Dateibaum angezeigt. Rechts unten werden Informationen zu dem im Dateibaum selektierten Verzeichnis bzw. der selektierten Datei angezeigt. Falls ein Datensatz Grafikdateien enthält, wird eine Vorschau angezeigt. Die Bilder lassen sich über die Schaltflächen mit den Pfeilen durchschauen. Über die *kopieren*-Schaltfläche wird das aktuell gewählte Bild in die Zwischenablage kopiert und kann in einer beliebigen Anwendung (z.B. Microsoft Word) eingefügt werden. Ein Doppelklick auf die Vorschaugrafik öffnet einen Dialog in dem die Grafik in Originalgröße angezeigt wird. Auf der linken Seite sind verfügbare Befehle aufgelistet.

4.3 Such-Ansicht

In der Such-Ansicht können die Metadaten aller Datensätze nach bestimmten Zeichenfolgen durchsucht werden.

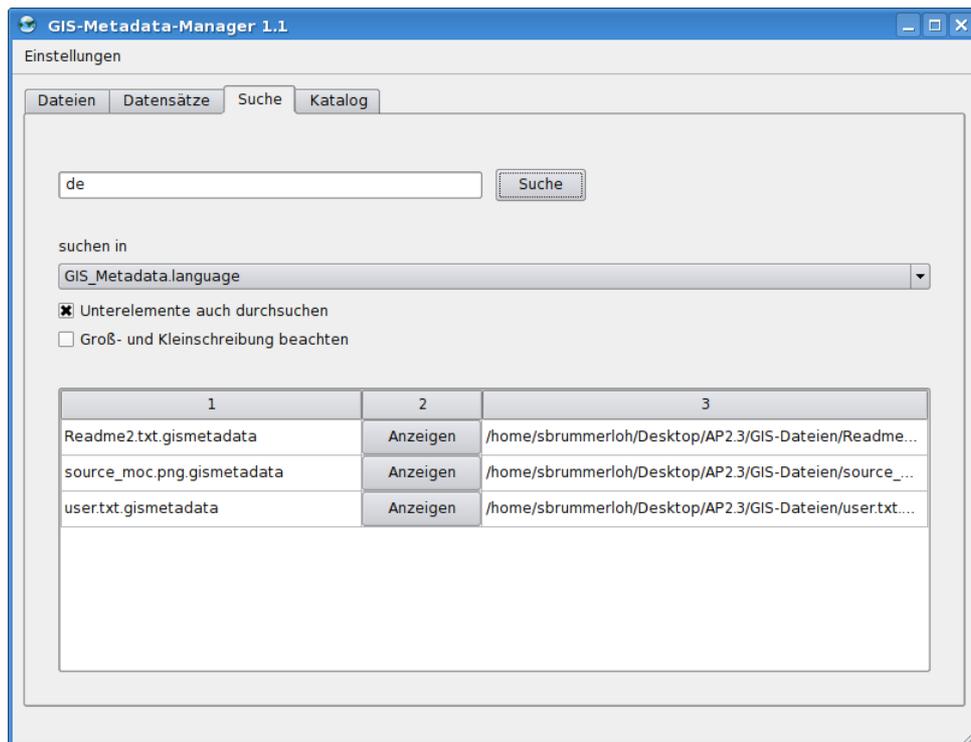


Abbildung 3: Die Such-Ansicht

Über *suchen in* können die zu durchsuchenden Metadatenelemente eingeschränkt werden. Wenn die Option *Unterelemente auch durchsuchen* aktiviert ist, werden alle Unterelemente des gewählten Metadatenelements mit durchsucht. Falls nur genau das gewählte Metadatenelement durchsucht werden soll, darf diese Option nicht markiert sein. Außerdem kann gewählt werden, ob die Groß- und Kleinschreibung des Suchbegriffs berücksichtigt werden soll. Suchergebnisse werden direkt in der Such-Ansicht angezeigt.

4.4 Katalog-Ansicht

Die Katalog-Ansicht ermöglicht es, schrittweise geeignete Datensätze zu ermitteln. Dazu können einzelne Ausprägungen von Metadatenelementen gewählt werden. Angezeigt werden dann jeweils nur die Datensatzdateien, die diese Ausprägungen besitzen.

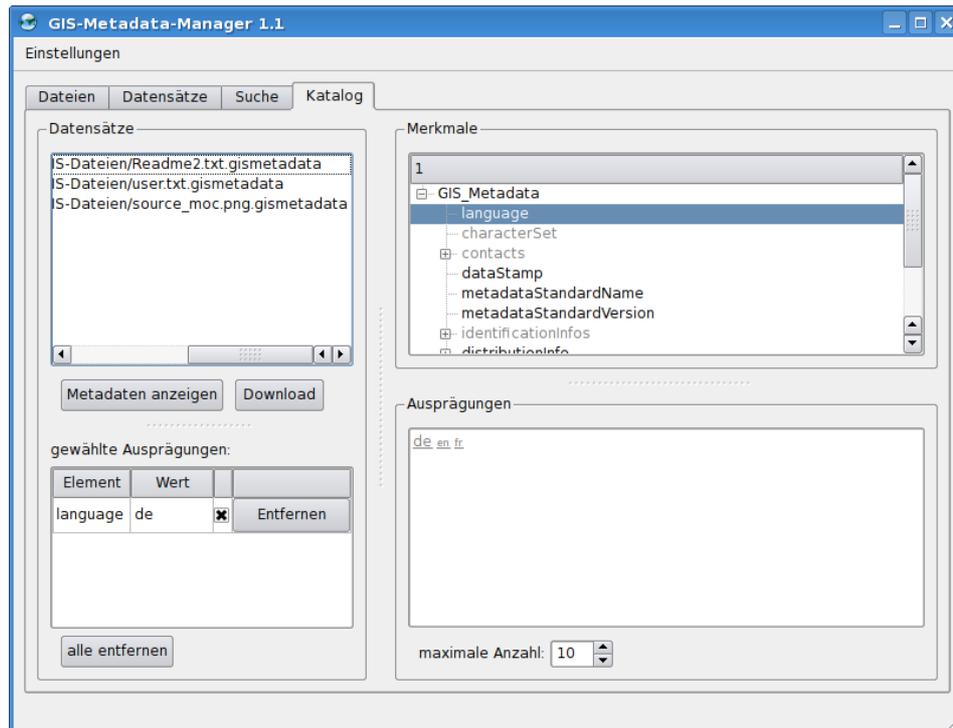


Abbildung 4: Die Katalog-Ansicht mit gewählter Ausprägung

Zum Erzeugen der Datenstruktur für die Katalog-Ansicht müssen zunächst alle Metadateidateien eingelesen und verarbeitet werden. Dies hat sich schon frühzeitig in der Entwicklung als sehr zeitaufwendig herausgestellt und den Startprozess des GIS-Metadaten-Managers um eine nicht akzeptable Zeit verzögert. Daher wurde eine Kommandozeilen-basierte Anwendung zum Analysieren des Verzeichnisbaums entwickelt. Diese prüft bei jeder Datei, ob es sich um eine Metadateidatei handelt. Zusätzlich werden aus den Metadateidateien die Metadaten ausgelesen und so aufbereitet, dass sie rasch von GIS-Metadaten-Manager für die Katalog-Ansicht verarbeitet werden können. Die Ergebnisse werden in einer XML-Datei gespeichert, die in den Einstellungen des GIS-Metadaten-Managers angegeben werden kann. Die Funktionalität der Kommandozeilen-basierten Anwendung zum Erzeugen der XML-Datei kann alternativ auch direkt in den Einstellungen des GIS-Metadaten-Managers aufgerufen werden.

4.5 Matadatendialog

Im Metadatendialog wird die Struktur der Metadatenelemente angezeigt. Es ist möglich, die Werte der einzelnen Elemente zu ändern, zu löschen und neue Elemente hinzuzufügen. Der Metadatendialog kann aus allen vier Ansichten des Hauptfensters aufgerufen werden.

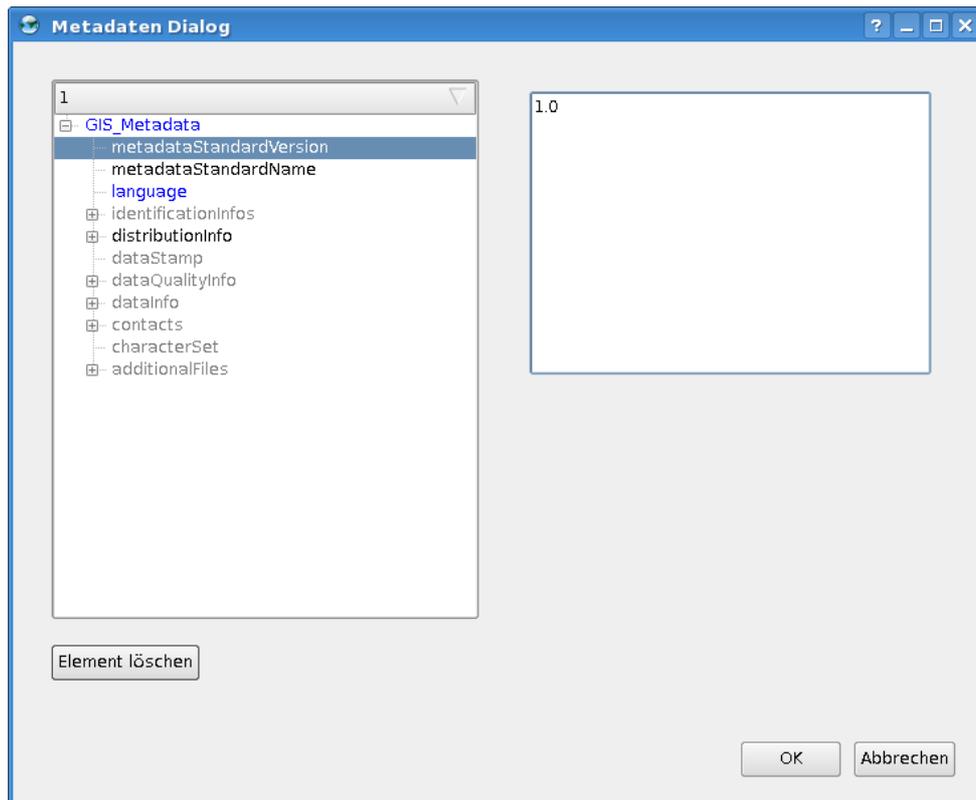


Abbildung 5: Der Metadatendialog zeigt die Metadaten zu einem Datensatz

Auf der linken Seite ist die Struktur der Metadatenelemente zu sehen. Auf der rechten Seite kann der Wert des Elements angegeben bzw. ein neues Unterelement zu dem selektierten Element hinzugefügt werden. Die Eingabefelder werden je nach Datenformat des Elements gewählt. Wenn ein Element ein Datum repräsentiert wird z. B. ein Kalender angezeigt.

5 Resultate und Nachhaltigkeit

Der GIS-Metadaten-Manager hat sich bei DLR-TT zum Verwalten von GIS-Dateien bewährt. Durch den Einsatz des GIS-Metadaten-Managers entstand der Wunsch nach weiterer Funktionalität, die bis zum Ende des WISENT-Projekts zu einem Teil integriert werden konnte. Der GIS-Metadaten-Manager wurde in dem Open-Source-Portal Sourceforge.net¹¹ als Open-Source-Projekt unter <http://sourceforge.net/projects/gis-manager> veröffentlicht und somit der Öffentlichkeit zur Nutzung und Weiterentwicklung zur Verfügung gestellt.

11 <http://sourceforge.net>