

# Automatische Generierung von OGC Styled Layer Descriptor Dateien (SLD) für das WebMapping auf Basis von ArcMap als visuellem Editor

Albrecht Weiser und Alexander Zipf

## Zusammenfassung

Eines der Ziele der vorliegenden Arbeit ist es den Umgang (insbesondere Konfiguration) mit internetbasierten Kartendiensten zu erleichtern. Ein hierzu entwickeltes neues Softwarewerkzeug "ArcGIS-map to SLD Converter" nimmt dem Benutzer die aufwendige Symbolisierung ab, indem es vorliegende ArcGIS-Karten auf ihre Symbole analysiert und die gewonnenen Daten in der OGC-Symbolisierungsvorschrift SLD (Styled Layer Descriptor) ausgibt. Diese Vorschrift kann auf entsprechenden OGC-konformen Kartenserver benutzt werden, um die analysierte Karte mit einem WMS im Internet zu veröffentlichen. Weiterhin wurden mit diversen Tests auf zwei unterschiedlichen WMS-Implementierungen die Möglichkeiten der aktuellen SLD-Spezifikation und deren Unterstützung durch die beiden getesteten WMS-Realisierungen (ArcIMS und UMN) erprobt. Diese sind zwar augenblicklich noch eingeschränkt, aber dennoch in der Praxis schon nutzbar.

Derartige SLDs von graphisch in Desktop GIS per Hand erstellter Karten können auch dazu verwendet werden, um sie mit weiteren Kontextparametern oder individuellen Benutzerinteressen zu versehen, um daraus Benutzer- und Kontext-, „adapative“ Kartenbeschreibungen zu generieren (Zipf 2003, 2005). Dies kann u.a. technisch durch eine Kaskade mehrerer XSLT-Transformationen geschehen. Merdes et al. (2005) präsentieren einen Ansatz, um aus SLD- und GML-Daten über eine entsprechende XSLT-Kaskade Karten als SVG-Dateien zu generieren. Dieser Ansatz wird durch die Nutzung von on XML kodierten User- und Kontextparametern, wie Sie Zipf und Jöst (2005) vorschlagen, durch eine Erweiterung der kaskade auch für personalisierte und kontextabhängige Kartenanwendungen nutzbar gemacht (Zipf 2005).

## 1 Einleitung

Zur Erstellung von Karten werden prinzipiell mindestens zwei Komponenten benötigt:

1. Eine Datengrundlage, welche die darzustellenden räumlichen Daten beinhaltet
2. Eine Abbildungsvorschrift, wie diese Daten auf einem optischen Endausgabegerät präsentiert werden sollen

SLD ist eine solche Abbildungsvorschrift für Karten, die von internetbasierten (und intranetbasierten) OGC-konformen Mapservern erzeugt werden. Die Symbolisierung ist eines der wichtigsten Elemente einer Karte, denn durch sie werden die Karteninhalte erst interpretierbar. Zu diesem Zweck wurde SLD von OGC entwickelt. Es soll für OGC-fähige Kartendienste die Darstellung der Karteninhalte nach o.g. Kriterien durchführen können.

Es soll also untersucht werden, ob SLD diese Kriterien mit den Möglichkeiten der derzeitigen Spezifikation erfüllen kann. Worin liegt der Sinn einer zusätzlichen Abbildungsvorschrift für Kartenserver, die (zumindest z. Zt.) weniger Darstellungsmöglichkeiten bietet als

die vorhandener proprietärer Systeme? Meistens bieten die Hersteller der Desktop-GIS-Systeme, die zur Erfassung und lokalen Bearbeitung der Daten von den Firmen, Kommunen und Behörden gekauft wurden, genügend Möglichkeiten um die Daten in entsprechender Qualität ins Internet oder Intranet zu bringen. Beispielsweise bieten ESRI und auch Intergraph (um nur zwei der marktführenden Unternehmen zu nennen) jeweils ein Produkt an, mit dem die Desktop-Projekte direkt im Internet veröffentlicht werden können. Außerdem gibt es von Drittanbietern Lösungen, um die Darstellungseigenschaften eines Desktop-Projekts zu extrahieren und als Kartendienst in einem proprietären Format abzuspeichern. Zu nennen wären z.B. AveiN! und AmeiN! der Firma Terrestris GbR, die ArcView- oder ArcMap-Projekte in UMN-Kartendienste umwandeln. Derartige Lösungen sind jedoch proprietäre Lösungen. WMS-SLD folgt dagegen einem internationalen Standard. Es gibt damit drei gute Gründe, warum man proprietäre Lösungen zugunsten von Standards, wie z.B. OGC, W3C oder ISO sie erarbeiten, vermeiden sollte:

### *1. Die allgemeine Nutzbarkeit und die Austauschbarkeit von Daten*

OGC SLD definiert eine gemeinsame Sprache zur Erstellung von Darstellungsanweisungen für Mapserver. Als Alternative besteht das Szenario, dass die Daten der Organisation X i.A. nicht auf dem Server der Organisation Y dargestellt werden können, weil beide Systeme von unterschiedlichen Anbietern und damit jeweils andere Schnittstellen nutzen.

### *2. Übertragbarkeit der Darstellungsvorschrift von einem Map-Server auf den anderen*

Jeder Map-Server benötigt eine Konfigurationsdatei, in der die zentralen Informationen enthalten sind, die notwendig sind, um einen Kartendienst zu errichten. Beim ArcIMS von ESRI beispielsweise heißt die Datei AXL-Datei (XML); beim UMN-Mapserver heißt sie Map-Datei (Text); beim Deegree-Server wiederum heißt die Datei wms\_capabilities (XML). In der Konfigurationsdatei sind Informationen über den Ort der Daten, Layernamen etc. enthalten. Aber auch die Symbolisierung der Klassen wird hier normalerweise in einem proprietären Format vorgenommen.

#### **Ein Rechenbeispiel:**

Nehmen wir ein Projekt wie die Geologische Übersichtskarte 1:300.000 (GÜK 300) als exemplarisches Projekt. Angenommen, die GÜK 300 ist aus 6 Layern zusammengesetzt. Jeder Layer enthält durchschnittlich 90 Klassen und für jede Klasse wird in der AXL-Datei des ArcIMS 6 Zeilen für eine Symboldefinition benötigt (ungefähr dieselbe Größenordnung wie in den Konfigurationsdateien bei UMN-Mapserver und Deegree auch). Das macht insgesamt  $6 \times 90 \times 6 = 3240$  Codezeilen, die händisch erstellt werden müssten, was natürlich sehr aufwendig ist. Nehmen wir weiterhin an, man möchte aus Kostengründen einen UMN-Mapserver als alternativen Kartenserver betreiben und darauf dieselbe Datenbasis als Kartendienst mit einbinden. Dann müsste eine zweite Konfigurationsdatei für den UMN-Mapserver mit dem gleichen Aufwand erstellt werden. Aber dieser Aufwand kann minimiert werden, wenn WMS-SLD als Darstellungsvorschrift benutzt wird, statt die komplette Symbolisierung in der Konfigurationsdatei zu erstellen. Das Aufsetzen einer Konfigurationsdatei wird einem dadurch zwar nicht erspart, aber der Umfang dieser Datei reduziert sich auf eine Größe von 20 bis 50 Zeilen (je nach Mapserver).

### *3. Die Kosten*

Proprietäre Technologien sind abhängig von Firmenpolitik und bieten somit wenig Planungssicherheit. Standards hingegen gewährleisten sichere Grundlagen für Entwicklungen. Somit lassen sich langfristige Entwicklungen planen und kostspielige Neuentwicklungen auf Basis einer veränderten Technologie vermeiden.

## 2 Die SLD-Spezifikation

Mit SLD bietet OGC die Grundlagen, die begrenzten Möglichkeiten eines gewöhnlichen WMS bezüglich der Darstellung von zu veröffentlichenden Daten wesentlich zu erweitern. Ein einfacher WMS (basic-WMS) benutzt zur Darstellung eine begrenzte Anzahl an vordefinierten Styles, die lediglich jeweils einem Layer zuweisbar sind. Komplexere Symbolisierungen sind auf diese Weise nicht möglich. Dafür steht dem WMS die Möglichkeit der Symbolisierung mit SLD zur Verfügung. Mit der SLD-Syntax können die Layer klassifiziert und den einzelnen Klassen unterschiedliche Symbolisierungen zugewiesen werden. Bei diesen sog. "SLD-WMS" können zusätzlich die Parameter „*DescribeLayer*, *GetLegendGraphic*, *GetStyles* und *PutStyles*“ eingebunden werden. Mit basic-WMS allein ist keinerlei Klassenbildung der vorhandenen Daten möglich - einem Layer wird ein einziger Style zugeordnet. Will man die Daten klassifizieren, d.h. attributabhängig in verschiedene Klassen einteilen und diese dann auch unterschiedlich darstellen, benötigt man ein weiteres Werkzeug, das OGC jetzt mit der SLD Spezifikation bietet. SLD basiert auf XML und liegt z.Zt. in der Version 1.0.0 mit der Spezifikation vom 19.08.2002 vor. SLD ist nicht nur in der Lage, mit WMS zusammenzuarbeiten, sondern kann auch benutzt werden, um Inhalte von WCS oder WFS darzustellen und entsprechend zu symbolisieren. Die SLD kann auf drei verschiedene Weisen in einen Dienst eingebunden werden:

1. mit einem http-GET Request "Inline“, wobei die SLD als Wert des CGI-Parameters SLD\_BODY übergeben wird:
2. mit einem http-GET Request, auf ein externes SLD-Dokument über eine URL als Wert des CGI-Parameters SLD:

```
http://mapserver:8080/WMServlet/guek300?request=getMap&srs=EPSG:31467
&bbox=3447309,5542994,3450522,5545762&layers=GEOLOGIE.GUEKm00_POLY&styles=
Style1&SLD=http://server/sld/SLD_valid.xml&format=png&width=250&height=400
```

3. mit einem http-POST Request, im "body"- Element einer aufrufenden Website in XML

Die 1. Methode kann leicht unübersichtlich werden und ist für sehr große SLD praktisch undurchführbar. Die 3. Methode ist technisch nicht einfach umzusetzen. Methode 2 ist ein guter Kompromiss zwischen 1 und 3 und wird in der Praxis am häufigsten verwendet. Eine interessante Möglichkeit, die SLD für einen WMS bietet, ist die Übernahme von Styles aus einer SLD in die Definitionsdatei des Mapservers. Wenn der Mapserver dies unterstützt, können mit dem CGI-Parameter "GetStyles" die in der Konfiguration des Mapservers definierten Styles abgefragt werden. Im Gegenzug können über den Parameter "PutStyles" Symboldefinitionen aus einer vorhandenen SLD in eine Mapserverkonfiguration übernommen werden. Dazu werden in der Aufruf-URL besagte Parameter eingeflochten und als Parameterwert der zu übernehmende Style aus der SLD gesetzt. Sowohl die *Layers* als auch die *Rules* innerhalb eines SLD-Layers sind "Z-ordered"; d.h. die Reihenfolge ihres Auftretens spielt für das Rendern eine Rolle.

### 2.1 Relevante SLD-Elemente

#### NamedLayer und UserLayer

*NamedLayer* und *UserLayer* sind Kindelemente des SLD Root-Element *StyledLayerDescriptor*. Ein *NamedLayer*-Element wird dann verwendet, wenn der Layername zuvor in der Konfigurationsdatei des WMS festgelegt wurde. Man kann den Layernamen überprüfen,

indem an den WMS eine GetCapabilities-Anfrage gestellt wird. In der auf die Anfrage folgenden Antwort sind alle zur Verfügung stehenden NamedLayer aufgelistet.

Ein *UserLayer* hingegen wurde noch nicht vorab in der Konfiguration des WMS definiert. Das Element dient dazu, per SLD aus einer vorhandenen Datengrundlage einen neuen Layer zu definieren. Allerdings wird dies z.Zt. noch von wenigen Mapservern unterstützt.

#### **NamedStyle und UserStyle**

Äquivalent zu den oberen beiden Elementen erfüllen die Präfixe "Named" und "User" der Elemente *NamedStyle* und *UserStyle* hier dieselben Aufgaben

#### **Rule**

Das *Rule*-Element ist sehr wichtig, da mit ihm die Definition einer Regel zur Klassifizierung des Layers eingeleitet wird. Alle notwendigen Informationen, sowohl zur Klassenbildung einer Klasse als auch deren Symbolisierung, stehen innerhalb des Elementes *Rule*.

#### **Filter**

Das Element *Filter* ist der "OGC Filter encoding Specification" entnommen. In ihr sind alle Termini definiert, die notwendig sind, um logische Operationen auszuführen, die der Funktion eines SQL-Statements gleichzusetzen sind. Da ein Layer ohne Klassifizierung einheitlich dargestellt würde, muss dem Mapserver mitgeteilt werden, nach welchen Bedingungen er die aktuelle Klasse erstellen soll. Diese Bedingung wird durch das Element *Filter* eingeleitet. Innerhalb des Filter-Elements können Vergleichsoperatoren wie *PropertyIsEqualTo*, *PropertyIsNotEqualTo*, *PropertyIsLessThan*, *PropertyIsGreaterThan* etc. sowie räumliche Operatoren wie *Touches*, *Overlaps*, *Intersects* u.v.a.m. als auch logische Operatoren wie *And*, *Or* und *Not* enthalten sein. Diese modellieren SFS-Funktionalität nach, so dass es mit dem Element *Filter* möglich ist, regelrechte Querys wie in SQL abzusetzen. Das Element *ElseFilter* kann in einer Klassifizierung dafür verwendet werden alle Werte, die mit Filter noch nicht klassifiziert wurden, einer alternativen Klasse zuzuweisen.

### **3 ArcMap und ArcObjects als visueller Editor zur Kartenerzeugung**

ArcMap bietet eine Fülle von Möglichkeiten zur graphischen Erstellung von Karten. Diese Informationen werden in einer proprietären binär-codierten Datei (\*.mxd) abgelegt. Mittels der Klassenbibliothek ArcObjects erhält man Zugriff auf die Eigenschaften einer Karte in einem gerade laufenden Projekt. Die Möglichkeiten sollen kurz charakterisiert werden. So bestehen diverse Symbolisierungsoptionen:

*Symboldefinitionen PointSymbolizer, LineSymbolizer, PolygonSymbolizer, TextSymbolizer, (RasterSymbolizer)*

Unter den jeweiligen Symboldefinitionen sind alle zur Symbolisierung der jeweiligen Klasse notwendigen Einstellungen und Eigenschaften festgelegt. Attribute wie Farbfüllung, Strich-, Text- und Punktfarbe, Strichstärke, Texthöhe, Punktgröße, Deckkraft der Farbe u.v.m., aber auch die Art von Flächenfüllungen (vollflächig, mit graphischen Füllelementen oder der Referenz auf ein hinterlegtes Bitmap) sowie das Aussehen von Linien (gestrichelt, gepunktet etc.) werden hier definiert. Eine der wichtigsten Funktionen in ArcMap ist die Möglichkeit zur Klassifizierung der Daten. Man findet hier fünf Hauptmethoden zur Klassifizierung:

1. *Features*: Die Daten werden alle als "Single Symbol" unklassifiziert mit der gleichen Symbolisierung dargestellt
2. *Categories*: Die können Daten nach einem oder maximal drei Datenbankfeldern klassifiziert werden, indem aus diesen Feldern die "Unique Values" gebildet werden.
3. *Quantities*: Hier werden die Daten aus einem bestimmten Feld in Wertebereiche eingeteilt und jedem Bereich eine bestimmte Symbolisierung zugewiesen.
4. *Charts*: Mit dieser Funktion können 0 bis maximal alle Felder einer Datenbank ausgewählt werden, um die darin enthaltenen Daten als Diagramm darzustellen. Es besteht die Möglichkeit, mehrere Diagrammartarten auszuwählen und damit jedes Objekt der Karte nach den Kriterien der ausgewählten Felder mit einem Diagramm zu versehen.
5. *Multiple Attributes*: Diese verbindet die Funktion von Categories und Quantities.

Es ist auch möglich Flächensymbole mit komplizierten Punktfüllungen und Schraffuren zu realisieren, die durch Hinterlegung von Bitmaps zusätzlich erweitert werden können.

*Punktsymbole* werden als Typen diverser Schriftarten (sog. ESRI TrueType Fonts) eingebunden. Somit ist es möglich, die Punktsymbolpalette beinahe beliebig zu erweitern.

*Flächenhafte* Symbole können grundsätzlich Liniensymbole und Punktsymbole enthalten. *Liniensymbole* wiederum können beliebig viele Punktsymbole enthalten (z.B. für sog. kartographische Linien, die eine unterschiedliche Sequenz versch. Punkt- und Strichsymbole enthalten können).

Diese Vielfalt der Möglichkeiten setzt ein entsprechendes umfangreiches Datenmodell voraus. Dieses Objektmodell wird durch die Klassenbibliothek "ArcObjects" von ESRI repräsentiert. Die Schnittstellen dieser Bibliothek legen die Funktionalität der Klassen offen. Damit wird das Setzen oder Abrufen von Eigenschaften einer Klasse sowie der Benutzung ihrer öffentlichen Methoden mittels einer .NET-fähigen Programmiersprache möglich. Gemäß den Klassifizierungsoptionen existieren in ArcMap folgende Renderer:

1. *SimpleRenderer* ≈ Single symbol (Features)
2. *UniqueValueRenderer* ≈ Unique values (Categories)
3. *ClassBreaksRenderer* ≈ Graduated colors, Graduated symbols (Quantities)
4. *ProportionalSymbolRenderer* ≈ Proportional symbols (Quantities)
5. *DotDensityRenderer* ≈ Dot density (Quantities)
6. *ChartRenderer* ≈ Charts
7. *BiUniqueValueRenderer* ≈ Quantity by Category (Multiple Attributes)

ArcObjects verfügt zudem über fünf Hauptsymboltypen mit diversen Untersymbolen (Text-Symbol, MarkerSymbol, LineSymbol, FillSymbol and 3DchartSymbol). MultiLayer-Symbole können eine beliebige Anzahl an Symbolen der gleichen Grundsymbolart beinhalten und jedes Symbol kann wiederum aus anderen Symbolen zusammengesetzt sein (Polygone z.B. können Linien und Marker enthalten).

## 4 Implementierung des SLD-Konverters

Der Konverter muss also selbstständig und flexibel auf die Zusammensetzung der oben dargestellten Symbole reagieren. Jedes dieser Symbole besitzt eine Reihe an unterschiedlichen spezifischen Eigenschaften und Parameter, die programmintern abgespeichert werden. D.h. es muss eine Speichermöglichkeit für eine unbestimmte Zahl an unterschiedlichen Symboltypen und deren Eigenschaften zur Verfügung gestellt werden. Es wurde angestrebt

alle Möglichkeiten, die mit SLD auf vektorbasierter Grundlage z.Zt. ausführbar sind umzusetzen. Das Programm erfüllt dabei folgende Aufgaben:

1. Es analysiert die Symbolisierung von ArcMap-Karten, speichert diese und entscheidet aufgrund der Analyse, welcher SLD-Code (welche Klassifizierung, welche Feature-Klasse) zu generieren ist, um das analysierte Projekt am treffendsten widerzuspiegeln.
2. Analyseseitig wird ein Overhead an Daten gespeichert, der vielleicht zukünftig benötigt wird, in SLD im Augenblick aber nicht abgebildet werden kann, weil es z.Zt. noch nicht die Möglichkeiten zur Umsetzung bietet. Damit trotzdem eine sinnvolle Ausgabe möglich wird, muss eine Generalisierung der Daten von dem komplexen ArcMap-Symbol zu dem einfachen SLD-Symbol stattfinden.
3. Die Daten müssen als XML-Datei entsprechend der SLD-Spezifikation erzeugt werden.

Der Konverter dient wie erwähnt zum Erstellen von SLD Darstellungsanweisungen aus Informationen, die durch eine Analyse eines vorliegenden ArcGIS-Projektes gewonnen wurden. Die Voraussetzung für die Nutzung der Applikation ist deshalb ein installiertes ArcGIS und eine laufende ArcMap-Session mit einem geöffneten Projekt. Die Applikation ist darauf ausgelegt, die anfallende Aufgabe automatisch und ohne viele Benutzerinteraktionen zu erfüllen.

- Alle in ArcMap vorhandenen Symbole werden in interne Strukturen ausgelesen.
- es wurden alle Renderer-Typen implementiert, die derzeit in SLD umsetzbar sind: UniqueValueRenderer (incl. many fields) / SimpleRenderer / ClassBreaksRenderer
- Noch nicht vollständig umgesetzt sind zur Zeit (August 2005): ChartRenderer / Dot-DensityRenderer / ProportionalSymbolRenderer / ScaleDependentRenderer

Folgende Darstellungseigenschaften sind für die einzelnen Featureklassen implementiert:

- Punkt-Feature: Farbe, Punktgröße, Drehwinkel
- Linien-Feature: Farbe, Linienstärke
- Polygon-Feature:
  - Einfarbige Flächenfüllung: Flächenfarbe, Linienfarbe und Linienstärke des Randes
  - Punktfüllung: Punktfarbe, Punktgröße, Linienfarbe und Linienstärke der Umrandung
  - Kreuzschraffur: Linienfarbe und Linienbreite der Schraffur, senkrechter und schräger Schraffurwinkel, Linienfarbe und Linienstärke der Umrandung

Ebenfalls implementiert ist die Darstellung von *Multilayersymbolen* als zweilagige Symbole (zweilagig, weil manche Mapserver z.Zt. nicht mehr Symbollagen prozessieren können). Für die *Beschriftung* der Symbole in einer Legende wurde für jede Regel (Rule) eines Symbols das Title-Element eingefügt. Als Elementwerte wurden die Klassen-Label der Symbol-Klassen aus der ArcMap-Karte benutzt. *Linienstrichlierung* sowie Bitmaps als Flächenfüllung sind derzeit (08/05) noch nicht realisiert. Der prinzipielle Ablauf der gesamten Nutzung des Programms wird in Abbildung 1 dargestellt.



**Abb 1.** Prozessdiagramm: Design einer Karte in ArcMap, Export derselben in ein SLD Document. Dieses kann genutzt werden, um WMS Server zu konfigurieren oder Nutzerspezifische Darstellungsanfragen per SLD an SLD-WMS zu senden.

## 5 Testen der SLDs und WMS-Kompatibilität

Es wurden verschiedene Tests bzgl. SLDs durchgeführt, die die Umsetzungsmöglichkeiten von Karteneigenschaften in SLD überprüfen. Damit sollen vor allem folgende vier Fragen beantwortet werden:

1. Wie gut unterstützen die eingesetzten Testserver z. Zt. die SLD-Spezifikation?
2. Welche Möglichkeiten der Darstellung von Karten gibt es grundsätzlich mit der gegenwärtigen Spezifikation? Das Hauptaugenmerk richtet sich dabei v.a. auf:
  - Symbolizer (Point-, Line-, PolygonSymbolizer)
  - Komplexe Flächenfüllungen wie Schraffuren und Punktfüllungen
  - Klassifizierungen mit Filter und logischen Operatoren
3. Welche Möglichkeiten der Erweiterung des Programms (zum Beispiel Flächenfüllungen mit Bitmaps) bieten sich an?
4. Welche Einstellungen müssen vorgenommen werden, damit die Testserver die SLD im Rahmen ihrer Möglichkeiten korrekt darstellen?

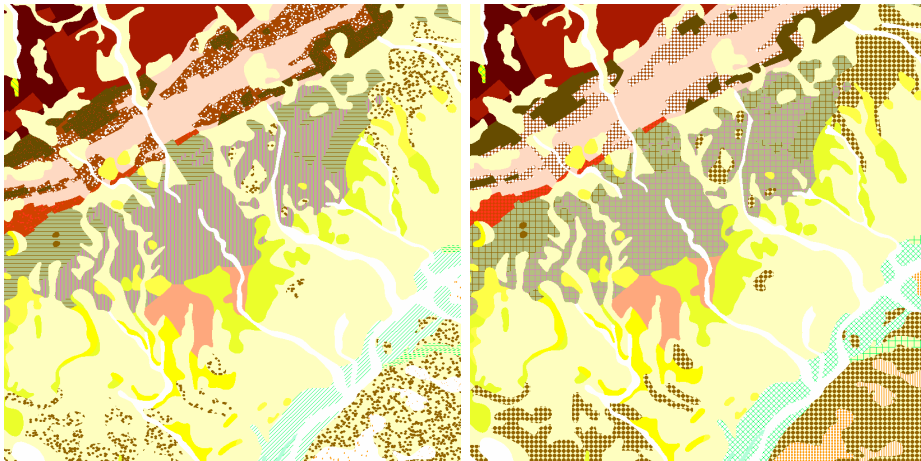
Die Tests wurden auf Basis eines realen Beispiels, der Geologischen Übersichtskarte 1:300.000 (GÜK 300) des Hessischen Landesamt für Umwelt und Geologie (HLUG) durchgeführt. Wer möchten hier noch einmal für die gute Zusammenarbeit mit dem HLUG danken. Bei den SLD-Funktionstests wurden zwei Testserver verwendet: ArcIMS von ESRI und der UMN-Mapserver, da diese weit verbreitet sind und zur Verfügung standen.

Die Elemente TextSymbolizer und RasterSymbolizer der aktuellen SLD-Spezifikation haben in dieser Arbeit keine Betrachtung gefunden, da der Fokus bei der Darstellung der

drei Featureklassen Punkt, Linie und Polygon lag. Ebenfalls keine Betrachtung fand die Generierung von Legenden zur Karte. Die Erzeugung von Kartenlegenden wird entweder clientseitig mittels JavaScript durchgeführt, oder serverseitig mit der *GetLegendGraphic*-Operation, die ebenfalls wie z.B. die *GetMap*-Operation einen WMS-Request darstellt. Im ersten Fall spielt SLD keine Rolle. Im Falle der serverseitigen Legendenerzeugung bietet SLD ein Element namens *LegendGraphic*, das jedoch laut [OGS] nur eine untergeordnete Rolle bei der Erzeugung von Legenden-Graphiken spielt.

Wie sich bei den Tests herausstellte, gibt es gravierende Unterschiede bei der Unterstützung von SLD-WMS durch die beiden Testserver. Die Tests konzentrierten sich auf die Darstellung der GÜK 300, weshalb dem Testen von Flächenfüllungen die größte Aufmerksamkeit zuteil wurde.

Bei den SLD-Tests hat der UMN-Testserver besser abgeschnitten als ArcIMS. Mit ihm sind komplexere Flächenfüllungen sowohl auf Vektorbasis als auch auf Rasterbasis möglich. Abbildung 2 zeigt die Umsetzung der Original-Karte aus ArcMap (links) mittels eines automatisch generierten SLD auf dem UMN WMS (rechts).



**Abb. 2.** Vergleich der Ergebniskarten (Ausschnitt) aus den generierten SLDs und dargestellt durch den UMN Mapserver (rechts) und die Originalkarte in ArcMap (links). Bsp.: Geologische Übersichtskarte 1:300.000, GÜK 300, Hessisches Landesamt für Umwelt und Geologie (HLUG).

Ein mögliches Kindelement des Elements *Mark* ist *WellKnownName*. *WellKnownNames* sind Elementar-Graphiken, die für WMS-kompatible Mapserver als vektorbasierte "Grundausstattung" mitgeliefert werden. Gemäß unserer Tests unterstützt UMN alle Elementar-Graphiken (als SVG) mit Ausnahme von *Square* and *Arrow*.

Der UMN-Mapserver ist jedoch nicht in der Lage mehr als zwei Symbollayer zu verarbeiten (*MultiLayer-Symbol*), da sonst ein Skriptfehler entsteht. Folgende Server-Fehlermeldung wird bei drei (oder >3) übereinanderliegenden Symbolen generiert. Der CssParameter *opacity* (Transparenz) wird für keines der Symbole vom UMN-Mapserver unterstützt. Zur Symbolisierung von Flächenfüllungen bieten derzeit Bitmaps die interessantesten Möglichkeiten. Es ergeben sich dadurch zwei Vorteile:



- die Möglichkeit zur "freien" Symbolwahl
- der Symbolabstand ist "frei" wählbar.

Im Gegensatz zu vektorbasierten Punktsymbolen, die "aneinanderstoßen" weil sie auf ihre Bounding-Boxes geclippt wurden, kann bei Bitmaps der Abstand des Symbols zum Symbolflächenrand variiert werden. Dies gilt jedoch strenggenommen nur für den UMN-Mapserver, da die Möglichkeit besteht, dass jeder Map-Server die Symbole etwas unterschiedlich darstellt, und auch Punktsymbole anders clippt als der UMN-Mapserver. Die Tabelle 1 im Anhang zeigt eine Übersicht der auf dem UMN-Mapserver getesteten Elemente und Attribute. Für eine komplette Auflistung aller SLD-Elemente und deren Unterstützung durch den UMN-Mapserver wird die offizielle UMN-Mapserver-Homepage empfohlen. Dort gibt es ein spezielles Kapitel zur SLD-Unterstützung (SLD HOWTO 4.6).

Alle für den UMN-Mapserver durchgeführten Tests wurden ebenso für den ArcIMS umgesetzt. Dabei sind wesentlich weniger SLD-Funktionen unterstützt als bei dem UMN-Mapserver. Die Tabelle 2 im Anhang zeigt die auf dem ArcIMS getesteten Elemente und Attribute. Sie stellt keine vollständige Übersicht der SLD-Tauglichkeit von ArcIMS dar, sondern konzentriert sich hauptsächlich auf die zur Darstellung der GÜK 300 wichtigen Elemente. Welche Konsequenzen hat diese Unterstützung für die Darstellung von Karten?

- Es können die Klassifizierungen "Simple Symbol", "Unique Values" und "Class Breaks" vorgenommen werden.
- Der Anzeigemaßstab kann begrenzt werden.
- Die Darstellung von Linien ist sehr begrenzt (keine Linienfarbe, keine Strichstärke, keine strichlierten Linien).

Die Elemente *Graphic* und *GraphicFill* sind für Punkte, Linien und Polygone nicht unterstützt. D.h. es können weder Füllmuster auf Vektorbasis noch auf Rasterbasis angezeigt werden. Lediglich vollfarbige Flächenfüllungen sind mit dem ArcIMS auf OGC-SLD-Basis derzeit möglich. Diese Aussage betrifft aber nicht die grundsätzlichen Funktionen von ArcIMS, sondern die Vollständigkeit der SLD-Unterstützung. Wird ArcIMS im „proprietären“ Modus betrieben sind durchaus komplexere Darstellungen möglich. So wurden Karten im ArcIMS einmal mit AXL und einmal mit XML symbolisiert. Beide wurden mit ArcIMS erzeugt und liefern dennoch wesentlich unterschiedliche Ergebnisse. Die logische Folgerung daraus ist, dass nicht ArcIMS dafür verantwortlich ist, dass so wenig der SLD-Funktionalität umgesetzt wird, sondern ein anderes Modul. Die Vermutung liegt nahe, dass der WMS-Connector, der benutzt wird, um den ArcIMS OGC-kompatibel zu machen, dafür verantwortlich zeichnet. Hier wird die Karteninformation von OGC-Syntax in AXL übersetzt. Durch diese Schnittstelle gehen wesentliche Informationen zur Kartendarstellung verloren. Wenn mit einer neuen Version des WMS-Connectors die SLD-Unterstützung ausgebaut werden sollte, könnten die Darstellungsmöglichkeiten von ArcIMS auch mit SLD ausgeschöpft werden.

## 5 Resümee

SLD ist noch eine junge Technologie. Aber die Möglichkeiten werden zukünftig immer vielfältiger, je weiter die Spezifikation ausgebaut wird. Wenn die Spezifikation mehr Möglichkeiten bietet, wird auch die Unterstützung von SLD-WMS durch Firmen und Organisationen, welche Kartenserver und internetbasierte GIS-Systeme anbieten, wachsen. Wenn die Technologie etabliert und umfangreiche Unterstützung der Syntax durch die Kartenser-

ver erreicht ist, kann es u.U. möglich sein, gänzlich auf die Erstellung proprietärer Konfigurationsdateien für Kartendienste zu verzichten. Die Möglichkeiten dazu sind derzeit mit dem SLD-Element *UserLayer* schon vorhanden. Über dessen Kindelement *RemoteOWS* (und zugehörige Parameter) können Geodaten eingebunden werden, ohne den Umweg über eine Konfigurationsdatei nehmen zu müssen.

Diese SLDs können auch dazu verwendet werden, um abhängig von Benutzer- und Kontextdaten „adaptive“ Karten über angepasste SLDs zu erzeugen (Zipf 2003, Zipf und Jöst 2005). Dies kann technisch z.B. durch eine Kaskade mehrerer XSLT-Transformationen erreicht werden, in denen die Nutzer- und Kontextinformationen aus anderen XML-Dateien mit den SLD-Dateien verbunden werden (Zipf 2005). Merdes et al. (2005) präsentieren einen ähnlichen Ansatz, um aus SLD und GML SVG-Karten zu erzeugen. Dieser Ansatz wird durch die Einbeziehung von User- und Kontextparametern, wie Sie von (2005) vorgeschlagen werden erweitert.

## 7 Literatur

- BURKE R. (2003): Getting to know ArcObjects. Hrsg. ESRI. USA: ESRI Press, 2003
- ORMSBY, NAPOLEON, BURKE, GROESS, FEASTER. (2004): Getting to know ArcGIS desktop. Hrsg. ESRI. USA: ESRI Press, 2004
- SLD HOWTO UMN-MAPSERVER: <http://mapserver.gis.umn.edu/doc46/sld-howto.html>
- MENG, L., ZIPF, A. and REICHENBACHER, T. (eds.) (2004): Map-based mobile services – Theories, Methods and Implementations. Springer Geosciences. Springer. Heidelberg.
- MERDES, M., J. HÄUBLER, A. ZIPF (2005): GML2GML: Generic and Interoperable Round-Trip Geodata Editing using Standards-Based Knowledge Sources. AGILE 2005. 8.th AGILE Conference on GIScience. Estoril. Portugal.
- ZIPF, A. (2002): User-Adaptive Maps for Location-Based Services (LBS) for Tourism. In: K. Woeber, A. Frew, M. Hitz (eds.), Proc. of the 9th Int. Conf. for Information and Communication Technologies in Tourism, ENTER 2002. Innsbruck, Austria. Springer Computer Science. Heidelberg, Berlin.
- OGC Inc.: Styled Layer Description Implementation Specification, Version 1.0.0, 2002(a).
- ZIPF, A and JÖST, M. (2005 in press): Implementing Adaptive Mobile GI Services based on Ontologies - Examples for pedestrian navigation support. In: CEUS - Computers, Environment and Urban Systems - An International Journal. Special Issue on LBS and UbiGIS. Pegamon Press, Elsevier.
- ZIPF, A. and HÄUBLER, J. (2004): An Evaluation of the OpenLS Specifications for multimodal mobile applications. In: The Journal of Geographic Information Sciences. CPGIS. Berkeley. California. Vol.10, No2. December 2004. pp. 117-127.
- ZIPF, A. (2003): Forschungsfragen für kontextadaptive personalisierte Kartographie. In: Kartographische Nachrichten (KN). Themenheft "Mobile Kartographie". 1/2003. Kirschbaum Verlag. S. 6-11.
- ZIPF, A. (2005): Using Styled Layer Descriptor (SLD) for the dynamic generation of user- and context-adaptive mobile maps - a technical Framework. 5th International Workshop on Web and Wireless Geographical Information Systems. December, 2005. Lausanne, Switzerland.

## 8 Anhang

**Tab. 1 Auf UMN 4.4.2, 4.6.0 beta2 bzw. ArcIMS 4.0.1 (mit OGC WMS Connector 1.0) bzgl. GÜK 300-Darstellung getestete Elemente/Attribute**

SLD-Element	Unterstützt UMN	Unterstützt ArcIMS	SLD-Attribut	Unterstützt UMN	Unterstützt ArcIMS
NamedLayer	ja	ja			
UserLayer	nein	nein			
NamedStyle	ja	ja			
UserStyle	ja	ja			
FeatureTypeStyle	ja	ja			
Rules	ja	ja			
Min-, MaxScaleDenominator	ja	ja			
Filter	ja	ja			
ElseFilter	ja	ja			
PropertyIsEqualTo	ja	ja			
PropertyIsBetween	ja	ja			
AND	ja	ja			
PointSymbolizer	ja	ja			
Fill	ja	nein	(im Folgenden Css-Parameter)		
			fill	ja	nein
			fill-opacity	nein	nein
Size	ja	nein			
Mark	ja	nein			
WellKnownName	ja	nein			
Rotation	nein	nein			
LineSymbolizer	ja	ja			
Stroke	ja	ja	(im Folgenden Css-Parameter)		
			stroke	ja	nein
			stroke-width	ja	nein
			stroke-opacity	nein	nein
			stroke-dasharray	ja	nein
GraphicStroke	ja	nein	(s. Stroke)		
PolygonSymbolizer	ja	ja			
Fill	ja	ja	(im Folgenden Css-Parameter)		
			fill	ja	ja
			fill-opacity	nein	nein
GraphicFill	ja	nein			
Graphic	ja	nein			
ExternalGraphic	ja	nein			