

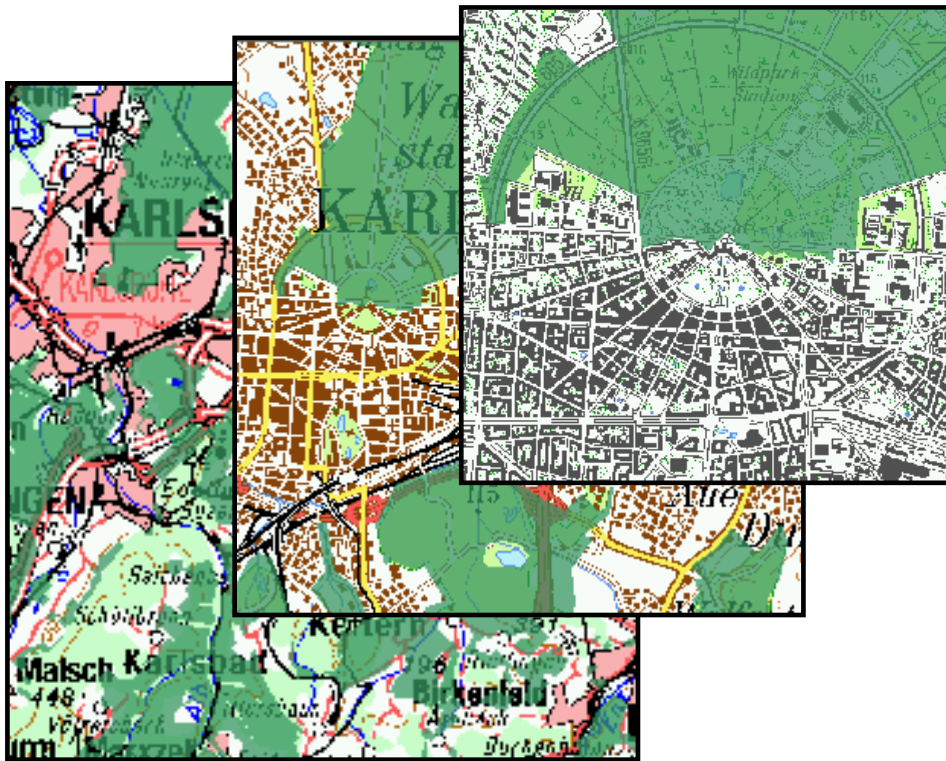


FACHHOCHSCHULE KARLSRUHE – HOCHSCHULE FÜR TECHNIK
Fachbereich Geoinformationswesen
Studiengang Kartographie und Geomatik

Diplomarbeit

von Tanja Schmidgall

unter der Leitung von Prof. Dr.-Ing. G. Schweinfurth



Konzeption und Entwicklung eines Verfahrens zur maßstabsgebundenen
Erzeugung und Abbildung von Umweltdaten

Landesanstalt für Umweltschutz
Baden-Württemberg



Diplomarbeit

für Frau Tanja Schmidgall

Thema: Konzeption und Entwicklung eines Verfahrens zur maßstabsgebundenen Erzeugung und Abbildung von Umweltdaten

Innerhalb des Umweltinformationssystems (UIS) Baden-Württemberg werden Datensätze bereitgehalten, die aus Karten unterschiedlichster Maßstäbe abgeleitet worden sind. Sollen nun Daten, die ursprünglich aus großmaßstäbigen Karten stammen, für eine landesweite Übersichtsdarstellung im kleinmaßstäbigen Bereich verwendet werden, ergeben sich zum einen graphische Probleme, zum anderen verbraucht ein hochauflöser, landesweiter Datensatz viel Speicherplatz und benötigt bei der Bildschirmvisualisierung lange Ladezeiten. Primär aus Performanzgründen ist es wünschenswert, detaillierte Daten für Übersichtsdarstellungen so aufzubereiten, dass ihr Speichervolumen reduziert und die Ladezeit beschleunigt wird.

Im Rahmen der Diplomarbeit soll deshalb ein Konzept entwickelt werden, wie großmaßstäbige Daten des UIS vereinfacht werden können, damit sie bei Verwendung im kleinmaßstäbigen Bereich dv-technisch gut handhabbar sind und darüber hinaus auch eine ansprechende Graphik liefern. Das Konzept ist programmtechnisch so umzusetzen, dass es mit Hilfe einer einfachen graphischen Benutzeroberfläche auch von einem Nicht-Geomatiker bedient werden kann. Das System ist mit ArcView 3.2 oder ArcGIS 8 unter Berücksichtigung der hier als Extension angebotenen Generalisierungsalgorithmen zu realisieren. Programmiersprachen sind Avenue bzw. Visual Basic.

Die vereinfachten Daten sollen innerhalb des landesweiten Intranet bzw. im Internet einem großen Nutzerkreis zur Verfügung stehen. Es ist deshalb noch zusätzlich zu recherchieren, welches der innerhalb der Landesanstalt für Umweltschutz (LfU) bereit gehaltenen GIS-Werkzeuge hierfür in Frage kommt.

Neben den entsprechenden Programmsystemen stehen für die Arbeit geeignete Datensätze (z.B. Landschaftsschutzgebiete) zur Verfügung. Entwicklungsumgebung ist Windows NT. Die Arbeit ist am ITZ der LfU Baden-Württemberg in Karlsruhe auszuführen.

Bearbeitungszeit: 6 Monate

Ausgabedatum:

Abgabedatum:

.....
Prof. Dr.-Ing. G. Schweinfurth

Für die Unterstützung und seine Betreuung meiner Diplomarbeit möchte ich Herrn Prof. Dr.-Ing. G. Schweinfurth herzlich danken. Bei technischen Problemen während meiner Arbeit mit den verschiedenen Programmen konnte ich mich jederzeit an Herrn Müller, Herrn Brass, Herrn Schillinger, Herrn Strittmatter und alle Mitarbeiter der LfU im Sachgebiet „Raumbezogene Informationssysteme“ wenden. Hier möchte ich allen für die gute Zusammenarbeit danken.

Für die nicht nur fachliche Hilfe beim Abschluss der Diplomarbeit bedanke ich mich bei meinen Eltern Beate und Manfred Schmidgall, sowie Jason Baker und Bianca Grams.

Ich widme die Diplomarbeit meinem Grossvater Willi Duschner. Ihm und meinen Eltern möchte ich für die Ermöglichung meines Studiums besonders danken.

Erklärung:

Hiermit versichere ich, dass ich diese Diplomarbeit selbständig verfasst und keine anderen als die von mir angegebenen Quellen und Hilfsmittel benutzt habe.

Karlsruhe, den _____ Tanja Schmidgall_____

Inhaltsverzeichnis:

	Seite
1 Die Einleitung	6
2 Das Projekt	8
2.1 Die Aufgabenstellung	8
2.2 Die Planung	8
2.3 Die Durchführung	9
3 Die Daten	10
3.1 Die Rasterdaten	10
3.2 Die Vektordaten	11
3.3 Raster-/Vektordaten: Gegenüberstellung	11
3.4 Die Geodaten	12
3.4.1 LSG und NSG	12
3.4.2 Die Kartengrundlagen	14
4 Die Programme	15
4.1 ESRI	15
4.1.1 ArcView	16
4.1.1.1 Spatial Analyst	17
4.1.2 ArcEditor	17
4.1.3 ArcInfo	18
4.1.3.1 ArcToolbox	18
4.1.3.2 ArcCatalog	18
4.1.3.3 ArcMap	18
4.2 disy	19
4.2.1 GISterm	19
4.2.2 MapServer	20
4.3 VB (VisualBasic)	20
4.3.1 VBA (VisualBasic for Applications)	20
4.3.2 VB Script	20
4.4 Der Douglas/Peucker-Algorithmus	21
5 Bearbeitung der Geodaten	22
5.1 Rastergeneralisierung	22
5.1.1 Rastergeneralisierung mit ArcView GIS	22
5.1.2 Rastergeneralisierung mit ArcInfo	27
5.2 Vektorgeneralisierung	32
5.2.1 Vektorgeneralisierung mit Arc GIS	32
5.2.2 Vektorgeneralisierung mit ArcInfo	34
5.3 Der Generalisierungsautomat	37
5.3.1 Erstellung der Benutzeroberfläche	37
5.3.2 Erläuterungen zum Macro	39
5.3.3 Anwendung des Automaten	39
5.3.4 Hilfe	42
5.3.5 Die Extension	43

6	Die Generalisierung im Test	45
6.1	Generalisierungsbeispiele	46
7	Die Interneteinbindung	47
7.1	Zusammenfassen der Daten mit GIStern	48
7.2	Der Internetaufruf	53
8	Zusammenfassung	56
9	Literaturverzeichnis	57
10	Abbildungsverzeichnis	58
11	Tabellenverzeichnis	59
12	Anhang	60

1 Die Einleitung

Der Bedarf an bodenkundlichen Informationen ist im Bereich des Umweltschutzes gravierend angestiegen. Bisher standen aktuelle Fachdaten nur den internen Bearbeitern zur Verfügung. Interessierte Nutzer hatten nur den Zugriff auf Kopien. Um Daten weitergeben zu können und dies auf dem neuesten Stand, werden mit Hilfe von Geoinformationssystemen (GIS) Daten gesammelt und weiter verarbeitet. Die Geoinformationssysteme übernehmen das Aggregieren, Analysieren, Simulieren und Modellieren der Daten, sowie Methoden der Kartographie. Die Originaldaten liegen bei der LfU auf einer ORACLE-Datenbank namens ITZGS1. Diese dient neben der Archivierung der Geodaten (RIPS-Pool) auch als zentraler Datenlieferant für alle ArcView-Anwender innerhalb der LfU.

Um die Daten vielfältig nutzen zu können, sollen sie schnellst möglich aufbereitet und nutzbar gemacht werden, was durch automatische Generalisierung ermöglicht werden soll.

„Forschung und Entwicklung zur rechnergestützten Generalisierung werden im Zusammenhang mit dem Aufbau und Einsatz von GIS intensiviert. Ziele sind hierbei die vielfältige Nutzung der digitalen Objektmodelle und die Herstellung kartographischer Darstellungen durch digitale Informationsverarbeitung.“ (HAKE S.337)

„Bereits Grundkarten stehen zur Realität in einem erheblichen Verkleinerungsverhältnis und erfordern daher bei den zu erfassenden Daten einen dem Maßstab entsprechenden Generalisierungsgrad. Folgekarten weisen gewöhnlich kleinere Maßstäbe auf als ihre Ausgangskarten. Würde man aber eine Ausgangskarte einfach nur photographisch verkleinern, so ergäbe sich bei fortgesetzter Anwendung dieses Vorgangs folgendes: Durch die Maßstabsverkleinerung schrumpft jede geometrisch exakte Wiedergabe eines Objektes immer mehr zusammen, bis sie schließlich unterhalb des Betrages der graphischen Mindestgröße liegen würde. Da dies aber nicht praktikabel ist, hat man wie folgt zu entscheiden:

- Man beachtet entweder das Prinzip der Lesbarkeit, muß dann aber das Objekt vergrößern, also unmaßstäblich wiedergeben und schränkt damit das Prinzip der geometrischen Richtigkeit ein oder

- man betreibt Verzicht auf Wiedergabe und schränkt damit jedoch das Prinzip der Vollständigkeit ein. Dieser Verzicht läßt sich entweder aus der geringeren Objektbedeutung oder aus Mangel an Darstellungsfläche begründen.“ (HAKE S. 110)

Diese beiden Prinzipien, sind also als Grundvoraussetzungen gegeben.

„Forschungen zur rechnergestützten Generalisierung sind in zweifacher Hinsicht notwendig: Erstens sind wirkungsvolle Verfahren für die automatisierte Ableitung von Datenmodellen geringerer geometrischer und semantischer Auflösung aus hochauflösenden Datenmodellen (Modellgeneralisierung zur Datenintegration) zu entwickeln. Zweitens sind die Forschungen zur kartographischen Generalisierung zu intensivieren, damit Verfahren entstehen, mit denen die Benutzeranforderungen an die Visualisierung digitaler Geo-Daten in flexibler Weise und mit guter Qualität erfüllt werden können[...].“ (HAKE S.335)

Des weiteren ist der Zeitgewinn eines Automaten nicht zu unterschätzen. Dies wirkt sich wiederum positiv auf die Aktualität des Karteninhaltes aus.

Um die Daten möglichst vielen Mitarbeitern innerhalb der LfU zur Verfügung stellen zu können, sollen sie in das Intranet bzw. Internet für die weitere Nutzung gestellt werden.

Das Intranet ist ein nicht öffentliches Landesnetzwerk, in dem alle relevanten internen und externen Informationsquellen in Form von HTML-Seiten bereitgestellt werden. Nahezu alle Daten lassen sich so für die Mitarbeiter in Form von digitalen Dashboards zur Verfügung stellen.

Vorteile des Intranets sind gemeinsam gepflegtes Firmenwissen, der aktuelle Stand der Arbeiten und umfassende Informationen schnell und ohne grossen Aufwand an den Mann zu bringen, von der Telefonliste bis zur Umweltkarte.

2 Das Projekt

2.1 Die Aufgabenstellung

„Konzeption und Entwicklung eines Verfahrens zur maßstabsgebundenen Erzeugung und Abbildung von Umweltdaten“

2.2 Die Planung

Zunächst war es nötig, eine klare Linie zu erarbeiten, die festlegt, welche Ziele wie erreicht werden sollen. Dafür wurden zahlreiche Gespräche mit Herrn Müller und seinen Mitarbeitern im Sachgebiet „Raumbezogene Informationssysteme“ geführt. Generalisiert werden sollen Umweltdaten, die als Punkte, Linien und Polygone bereits vorliegen. Als Generalisierungsobjekte sind die Naturschutzgebiete, sowie die Landschaftsschutzgebiete anzusehen.

Die Generalisierungsmaßstäbe wurden mit den bereits innerhalb der LfU eingeführten Maßstäben (M1-M3) festgelegt, wobei eine Änderung, auch in kleinere Maßstäbe denkbar war.

Maßstab	von	bis	für
M1	1:1	< 1:10.000	ALK
M2	1:10.000	< 1:50.000	ATKIS
M3	1:50.000	< 1:750.000	UDK
M4	1:750.000	< 1:1,5 Mio.	UDK
M5	1:1,5 Mio.	< 1: ~	UDK

Tabelle 1: mögliche Maßstabsbereiche

M4 und M5 sollten für den UDK (Umweltdatenkatalog) hergestellt werden, wobei hier die genaue vorherige Festlegung der Maßstäbe nicht möglich war und mit Werten, die es zu testen galt, angegeben wurde.

Mögliche Lösungswege für die Generalisierung wurden in der Avenue-Programmierung gesehen, die es innerhalb von ArcView GIS ermöglicht, Skripte zu schreiben.

Zur Präsentation im Intranet wurden als mögliche Programme ArcView, ImageMapper, der MapServer und auch GISterm angesehen. Hier galt es zu recherchieren, welche der Möglichkeiten zum gewünschten Ziel führen.

Die Generalisierung sollte so programmiert sein, dass es wenige „Handgriffe“ benötigt, um sie durchzuführen. Wünschenswert wäre eine Oberfläche, bei der der Faktor der Generalisierung über einen Schieberegler beeinflusst wird. Das Endprodukt der Generalisierung soll mit Kartenhintergrund der jeweiligen TK oder TÜK im Intranet veröffentlicht werden.

2.3 Die Durchführung

Die Durchführung gestaltete sich schwieriger als anfangs erwartet. Zuerst mussten die LfU-internen Abläufe erkundet und die benötigten Programme erlernt werden. Ein bereits vorhandener Algorithmus von Douglas/Peucker (1973) schien von großem Vorteil. Um diesen zu verstehen und damit arbeiten zu können, musste die Programmiersprache Avenue innerhalb von ArcView GIS erlernt werden. Dies konnte auch später genutzt werden, um Skripte in ArcView zu schreiben und diverse Ideen auszuprobieren.

Der Douglas/Peucker-Algorithmus erwies sich als zu ungenau für große Maßstäbe, so dass noch andere Möglichkeiten gefunden und weitere Lösungen erstellt werden mussten. Außerdem sollten auch Grids bearbeitet werden, was mit Douglas/Peucker nicht möglich war. Bei Internetrecherchen wurden Erweiterungen für ArcView gefunden wie z.B. Scripte für den Spatial Analyst, der als Extension in ArcView angehängt wird, damit in ArcView mit Grids gearbeitet werden kann.

Eine kurze Exkursion über Access und Smallworld erwies sich als unbrauchbar. Diese Programme sind hauptsächlich für Tabellenkalkulation und –manipulation nutzbar, doch für diese Zwecke unzureichend.

Als sehr nützlich erwies sich ArcInfo. Zu den bereits in den Vorlesungen erlernten Fähigkeiten, mussten einige Ergänzungen erlernt und getestet werden. Die anfänglichen Schwierigkeiten legten sich schnell und ArcInfo stellte sich als äußerst brauchbare Komponente für die Generalisierung heraus.

Um die generalisierten Umweltdaten darstellen zu können, fehlte nun noch eine Graphikkomponente. Diese wurde durch das Geodatenzugriffssystem GIStern gestellt. Es ermöglicht den Zugriff auf räumlich verteilte Geodaten, um aus den verschiedenen Quellen eine Karte zu kreieren. Diese Karten werden als sog. mml-Dateien abgespeichert und können so interaktiv exportiert oder zu einer thematischen Karte weiterverarbeitet werden.

Diese Eigenschaft wurde dann genutzt, um sie im Intranet mit dem MapServer darzustellen.

Innerhalb ArcInfos waren mehrere Schritte nötig, um die Daten zu generalisieren und für die Weiterverarbeitung vorzubereiten. Dies war nicht in meinem Sinne und ich musste mich nach anderen Möglichkeiten umsehen, um möglichst einen Automaten zu erstellen, der alle Schritte komprimierte. Hierfür musste der Umgang mit ArcCatalog und ArcToolbox erlernt werden. Mit Hilfe der beiden Programme, die eine Erweiterung von ArcInfo darstellen, war es möglich die Generalisierung durchzuführen, aber auch hier war es nötig, mehrere Schritte zu gehen. Als letzte Lösung erschien das Schreiben eines Makros. Um Makros für diese Problematik, unabhängig von Programmen schreiben zu können, musste die Programmiersprache Visual Basic (VB) erlernt werden.

Mit einigen Startschwierigkeiten wurde das Ziel erreicht, in dem innerhalb eines Makros alle Befehle vereint wurden, die für die Generalisierung von Nöten waren. So konnte ich die Daten mit dem MapServer ins Intranet stellen und den Mitarbeitern eine schnelle Möglichkeit bieten, um Karten zu vereinfachen und mit geringerem Speicherbedarf weiterzuverarbeiten.

3 Die Daten

3.1 Die Rasterdaten

Die Flächen von Rasterdaten werden in quadratische Zellen eingeteilt. Möglich sind auch Rechtecke, Matrizen oder Trapeze, die bei geographischen Koordinaten als Bezugssystem dienen. Die Topologie ist gegeben, das heißt die Nachbarschaft der Zellen, ist durch den Spalten- und Zeilenindex vorgegeben. Die geometrische Form wird verändert, so dass die Dimensionen von Punkt und Linien zu Flächen werden oder gänzlich verloren gehen.

Rasterdaten werden in ArcView als Grids bezeichnet.

Die einzelnen Zellen sind in einer Tile-Block-Cell-Struktur hierarchisch geordnet. Sie werden als separate Files gespeichert. Es ist meist nicht mehr als ein Tile erforderlich, da ihre maximale Größe sehr hoch ist. Tiles werden weiter unterteilt in Blocks. In den Blocks sind die Zellenwerte über den Zeilen- und Spaltenindex gespeichert. Wird ein Wert abgefragt, so wird zuerst das Tile, dann der Block und danach der dazugehörige Zeilen- und Spaltenindex herausgelesen. Die Griddaten werden in Blöcken zusammengefasst. Serien gleicher Werte werden durch Anfangs- und Endkolonne sowie dem Wert selbst beschrieben. Es wird zusätzlich überprüft, ob dadurch ein Platzgewinn entsteht. Ist das nicht der Fall, so wird jede Zeile einzeln gespeichert. Diese Struktur und die Zusammenfassung in Blöcke ist für den Nutzer nicht sichtbar.

Rastermethoden sind dann von Vorteil, wenn für jeden beliebigen Punkt eine Information vorhanden und schnell abrufbar sein muss.

Somit können beliebige Größen der Grids bearbeitet werden. Vom Analysefenster ist die Rechnerzeit abhängig, nicht von der Größe des Grids. Die maximale Größe ist abhängig vom Speicherplatz.

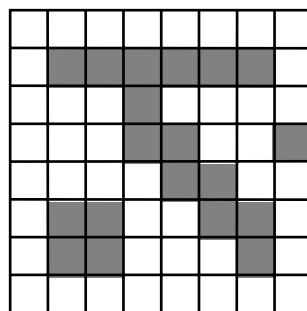


Abb. 1: Rasterformat

3.2 Die Vektordaten

Vektordaten sind Geometrien, die zur Darstellung linearer Merkmale mit Hilfe von Objekten und Koordinaten verwendet werden. Die Form eines Merkmals wird durch mehrere Koordinaten definiert. Die Koordinaten werden im einzelnen als Nodes oder Knoten bezeichnet. Die x-, y-Koordinaten sind in einer Liste geordnet und werden vom Objekt repräsentiert. Die einzelnen Attribute sind mit dem Objekt verbunden. Vektordaten werden in ArcView in Form von Shapefiles gespeichert.

Im GIS bestehen die Vektorformate aus Punkten (Knoten) und Verbindungslinien, den Vektoren. Sind Anfangs- und Endkoordinaten die selben, so ergibt sich eine Vektorfläche. Durch die Knoten ist die Vektorfläche im Raum lokalisierbar, wobei sie keinen Auflösungseinschränkungen unterworfen ist. Problematisch ist eine genaue Formbeschreibung, da Bogen und Kurven durch zusammengesetzte Geraden ersetzt werden. Manche GIS besitzen hierfür Splines zur Kurvenerzeugung. GIS unterscheiden Geometrie und Topologie. Die Geometrie gibt im Raum die Position des Objektes an und die Topologie die Lage der Objekte zueinander. Topologien werden von hoch entwickelten Vektordatenmodellen mit eingeschlossen. Im GIS werden Topologien berechnet und gespeichert.

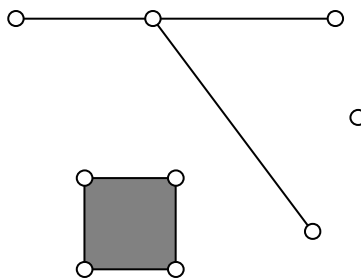


Abb. 2: Vektorformat

3.3 Raster-/ Vektordaten: Gegenüberstellung

Im Vergleich der beiden Möglichkeiten, scheint das Rasterformat von Nachteil zu sein. Es erscheint ungenau und einen größeren Speicherbedarf zu besitzen. Wenn man dies über große Bereiche annimmt, ist dies als richtig zu erachten. Sind allerdings Werte vorhanden, die sich von Ort zu Ort gleichmäßig verändern, werden nur die Koordinaten der Eckpunkte und die Art des Rasters gespeichert. Deshalb können sie in diesem Fall weniger Speicherplatz benötigen als die Vektordaten. Da für die Weiterverarbeitung das Shapeformat von Nöten ist und dieses für Vektordaten konzipiert ist, wird das Rasterformat nachteilig, da für jeden Rastereckpunkt ein Punkt gespeichert wird. Hier ist dann wieder das Vektorformat vorzuziehen, da es umgewandelt in das Shapeformat nur an den Kanten und an den Kurven Punkte setzt, also an den Stellen, an denen die Richtung sich ändert.

3.4 Die Geodaten

Die Daten bilden den Kern aller Geoinformationssysteme (GIS), sie beschreiben reale und abstrakte Objekte unserer Umwelt und sind in einer Datenbank zusammengefasst. Sie geben die strukturellen, topographischen und thematischen Eigenschaften der Objekte wieder und werden Geodaten oder Umweltdaten genannt. Der größte Teil der Geodaten wird aus Plänen digitalisiert oder gescannt und danach nochmals überarbeitet. Ein Problem stellt die dadurch bereits eingetragene kartographische Bearbeitung, die für einen bestimmten Zweck gemacht wurde und deren Genauigkeit für die aktuellen Umstände eventuell bei der Erfassung nicht ausreichend beachtet wurde.

Um die Geodaten bearbeiten zu können, müssen für ein komplettes GIS eine Reihe von Werkzeugen (Programmen) zur Verfügung stehen.

Geodaten, die im Internet veröffentlicht werden, wird teils unberechtigterweise ein großes Maß an Aktualität zugeschrieben. Um dem entgegenzuwirken, sollte darauf geachtet werden, dass immer der neueste Stand dem Nutzer zur Verfügung steht.

3.4.1 LSG und NSG

Die Umweltdaten werden bei der LfU in einer ORACLE-Datenbank gehalten. Diese können in das Shapeformat konvertiert werden.

Die Daten enthalten die digitalen Grenzen der Natur- und Landschaftsschutzgebiete sowie der Naturparks im Maßstab 1:25.000 mit Stand 31.12.1997. Diese Bodenübersichtskarten entstehen durch die Zusammenfassung von Bodeneinheiten, in dem kleinere Flächeneinheiten zu größeren Arealen werden. Diese Zusammenfassung nennt man auch Aggregation.

Bei der Nutzung der Daten ist immer darauf zu achten, dass für jedes Schutzgebiet das jüngste Verordnungsdatum (Vo_datum) verwendet wird. Dies trägt zur Aktualhaltung der überarbeiteten Daten bei.

Zu jedem Gebiet gibt es Angaben zu Lage, zu Größe und teilweise auch mit Fotos gezeigte Ausschnitte.

Landschaftsschutzgebiete (LSG): Land Baden-Württemberg und Regierungsbezirke:

Stand: 31.12.1999

Regierungsbezirk	Anzahl	Fläche in ha	% der Fläche ¹⁾
Stuttgart	584	236.112	22,4%
Karlsruhe	291	173.437	25,1%
Freiburg	244	169.352* ⁾	18,1%* ⁾
Tübingen	389	197.956	22,2%
Baden-Württemberg	1508	776.857*⁾	21,7%*⁾

1) 9 NSG erstrecken sich über 2 Regierungsbezirke, woraus sich die Differenz zur Summe in den Regierungsbezirken ergibt

*⁾ inklusive gemeindefreies Gebiet Rhinau

% berechnet nach den Angaben im Statistischen Taschenbuch Baden-Württemberg 1999 Stand: 31.12.1997.

Tabelle 2: Landschaftsschutzgebiete

Naturschutzgebiete (NSG): Land Baden-Württemberg und Regierungsbezirke:

Stand 31.12.1999

Regierungsbezirk	Anzahl	Fläche in ha	% der Fläche
Stuttgart	222	12081,5	1,14%
Karlsruhe	213	18672,5	2,70%
Freiburg	237	27180,6	2,90%
Tübingen	282	17495,3	1,96%
Baden-Württemberg	945¹⁾	75429,9	2,11%

% berechnet nach den Angaben im Statistischen Taschenbuch Baden-Württemberg 1999 Stand: 31.12.1997.

¹⁾ 9 NSG erstrecken sich über 2 Regierungsbezirke, woraus sich die Differenz zur Summe in den Regierungsbezirken ergibt.

Tabelle 3: Naturschutzgebiete

Natur- und Landschaftsschutzgebiete: Daten

Natur und Landschaft	Verzeichnisname	Dateiname	Format
Moorkataster (1:5.000 / 1:25.000)	natur_landschaft \moore	moore	Shapefile
Moorkataster - Bohrpunkte (1:5.000 / 1:25.000)	natur_landschaft \moore	bohrpkt	Shapefile
Bannwälder	natur_landschaft	bann_m1	Shapefile
Landschaftsökologischer Atlas 1:200.000	natur_landschaft	landsoek200	Shapefile
Landschaftsschutzgebiete 1:25.000	natur_landschaft	lsg25	Shapefile
Landschaftsschutzgebiete 1:250.000	natur_landschaft	lsg250	Shapefile
Landschaftsschutzgebiete 1:250.000 (<20 ha)	natur_landschaft	lsg250p	Shapefile
Naturparks 1:25.000	natur_landschaft	nap25	Shapefile
Naturschutzgebiete 1:25.000	natur_landschaft	nsg25	Shapefile
Naturräumliche Gliederung 1:200.000	natur_landschaft	nag200	Shapefile
Naturschutzgebiete 1:250.000	natur_landschaft	nsg250	Shapefile
Naturschutzgebiete 1:250.000 (<20 ha)	natur_landschaft	nsg250p	Shapefile
Schonwälder	natur_landschaft	schon_m1	Shapefile
Landschaftszerschneidung 1:25.000	natur_landschaft	uzrbwg25	Shapefile

Tabelle 4: Daten

3.4.2 Die Kartengrundlagen

Die digitalen Topographischen Karten sind Rasterdaten, die durch Scannen der originalen Kartenblättern oder durch Digitalisieren der topographischen Kartenwerke erzeugt wurden. Die Gesamtheit aller Karten nennt man Geobasisdaten. Diese Daten in Form von Raster- und Vektordaten bilden eine räumliche Bezugsgrundlage und sind im einzelnen eine wichtige Voraussetzung für die Kartierung der geothematischen Sachverhalte und Detailinformationen. Die Geobasisdaten sind in verschiedenen Maßstabsbereichen von der TK25 bis zur TÜK1000 und teilweise einfarbig und farbig vorhanden. Somit sind sie in flexibler und vielfältiger Weise in verschiedenen Fachbereichen anwendbar. Vor allem die Rasterdaten der TKs eignen sich als Hintergrundinformation für eine Überlagerung mit den spezifischen Themen.

Topographische Karten:

Topographische Karten	Verzeichnisname	Dateiname	Format
TK 1:25.000 (Farbe)	topkarten\tk25	tk6221co - tk8423co	TIF
TK 1:25.000 (Graustufen)	topkarten\tk25\einfa rbig	tk6221sw - tk8423sw	TIF
TK 1:50.000 (Farbe)	topkarten\tk50	L6316 - L8522	TIF
TK 1:50.000 (Graustufen)	topkarten\tk50\einfa rbig	L6316sw - L8522sw	TIF
TK 1:100.000 (Farbe)	topkarten\tk100	c6314 - c8722	TIF
TK 1:100.000 (Graustufen)	topkarten\tk100\einfa rbig	c6314sw - c8722sw	TIF
TÜK 1:200.000 (Farbe)	topkarten\tk200	cc6310co - cc8726co	TIF
TÜK 1:200.000 (Graustufen)	topkarten\tk200\einfa rbig	cc6310sw - cc8726sw	TIF
TÜK 1:200.000 (SW) - ges. Ba-Wü	topkarten\tk200\einfa rbig	tuek200sw	TIF
Übersichtskarte 1:500.000 (Farbe)	topkarten\tk500	Uk500co	TIF
Übersichtskarte 1:500.000 (Farbe, Höhenstufen)	topkarten\tk500	Uk500co_hs	TIF
Übersichtskarte 1:500.000 (Graustufen)	topkarten\tk500	Uk500sw	TIF
Übersichtskarte 1:1.000.000 (Farbe)	topkarten\tk1000	Uk1000co	TIF
Übersichtskarte 1:1.000.000 (Graustufen)	topkarten\tk1000	Uk1000sw	TIF

Tabelle 5: Topographische Kartengrundlagen

4 Die Programme

Innerhalb meiner Diplomarbeit kam ich mit einer Reihe von Programmen in Berührung. Den größten Teil der Arbeit konnte ich mit Hilfe der Produkte von ESRI und disy erreichen, die im folgenden beschrieben werden. Die Nutzung und den Umgang mit den Microsoftstandards wie z.B. Word und Excel setze ich hierbei voraus.

4.1 ESRI



ESRI ist die weltweit führende GIS-Company. Sie wurde 1969 durch den heutigen Präsidenten Jack Dangermond gegründet und blieb stets im Laufe ihrer Existenz mit klarer Kundenausrichtung. Hauptsitz von der ESRI-Company ist in den Redlands, Kalifornien, in den Vereinigten Staaten. Sie hat mehr als 1200 Mitarbeiter und agiert in mehr als 229 Ländern. Für alle Aufgabenfelder raumbezogener Informationsverarbeitung bietet ESRI GIS-Produkte an. Die Produkte von ESRI lassen sich unter dem Namen ArcGIS zusammenfassen.

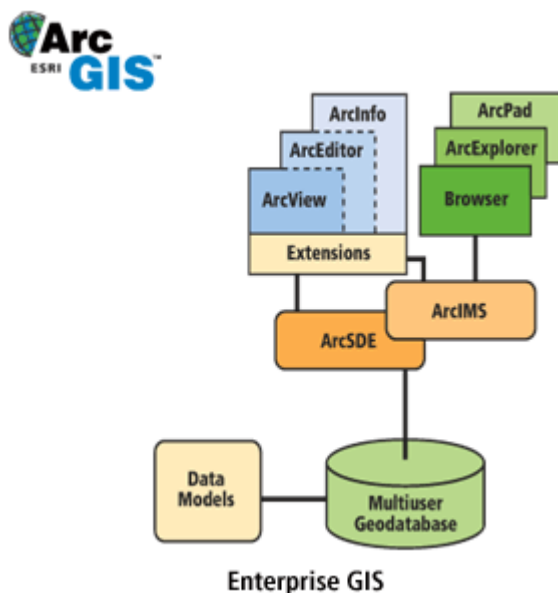


Abb. 3: ArcGIS

Alle Daten sollen in einer Datenbank gesammelt und gespeichert werden. Innerhalb von ArcGIS gibt es Server-Dienste und Clienten. Die Server-Dienste sind ArcSDE ein Datenbankgateway und ArcIMS, ein Internet Map Server. Sie stellen die Grundfunktionalitäten dar.

Die Clienten sind ArcView, ArcEditor und ArcInfo, dies sind voneinander abgestufte Anwendungen. Es gibt noch weitere Clienten wie ArcExplorer, ArcReader und ArcPad. Diese sind speziell für browserbasierte Anwendungen, die von HTML- oder Java unterstützt werden.

Die wichtigen Clienten sind:

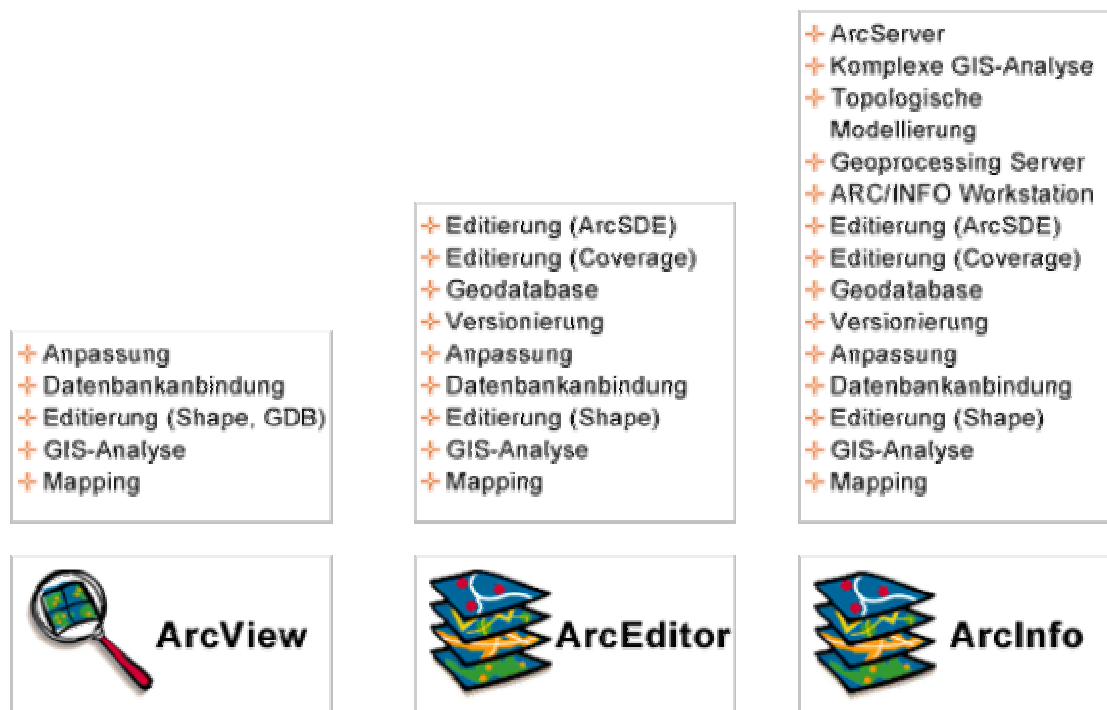


Abb. 4: ESRI

4.1.1 ArcView

Anfangs war ArcView ausschließlich zur Visualisierung raumbezogener Daten entwickelt worden. Seither ist die Software soweit vorangebracht worden, dass ein leistungsstarkes GIS-System entstand. Dieses bietet nun auch die Möglichkeiten, neben der Visualisierung, die Erfassung, die Abfrage, die Manipulation und die Analyse der raumbezogenen Daten. Um den Funktionsumfang erweitern zu können, bietet ArcView GIS die objektorientierte Programmiersprache und Entwicklungsumgebung Avenue an. Diese Sprache ist leicht zu erlernen. Innerhalb der ESRI-Homepage werden Avenue-Scripte von Anwendern angeboten, die sich fast täglich erweitern.

4.1.1.1 Spatial Analyst

Um in ArcView mit Grids arbeiten zu können, muss die Spatial Analyst Extension aktiviert sein. Dies kann im Extension-Manager eingestellt werden. Der Spatial Analyst speichert Grids entweder als integer oder als floating-point Grids. Integer verfügen über eine Valueattribut Tabelle (VAT). Zellen, die gleiche Werte besitzen, werden zu einer Zone zusammengefasst, welche dann in der VAT erscheint. In der Grundfunktion besitzt eine VAT die zwei Attribute Value und Count. Der Wert von jeder Zone wird in Value gespeichert zusammen mit der Anzahl der Zellen, welche zu dieser Zone gehört. Falls mehr als 500 verschiedene Werte vorhanden sind, wird auf die Bildung einer VAT verzichtet, dies kann aber mit dem Befehl BUILDVAT erzwungen werden.

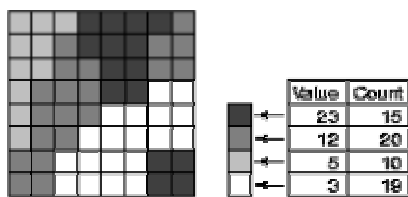


Abb. 5: Value und Count

Die Bildung einer VAT macht bei floating-point Grids keinen Sinn, da theoretisch jede Zelle einen anderen Wert haben kann. Für jede Zelle ohne Wertangabe, wird der Nodata-Wert verwendet. Der Nodata-Wert ist nicht 0!

In einem aktiven View können Grids hinzugefügt werden. Die Darstellung von Grids kann mittels des Legenden Editors verändert werden. Die Darstellung von Nodata-Werten in der Legende kann unterdrückt werden.

4.1.2 ArcEditor

ArcEditor stellt alle nötigen Arbeitsschritte für Editierarbeiten zur Verfügung. Es stellt Hilfen für die Erstellung, Pflege und das Management von großen Datenbeständen.

4.1.3 ArcInfo

ArcInfo stellt das kompletteste GIS dar. Es beinhaltet die Funktionen von ArcView und ArcEditor und ist durch Geoprocessing- und Datenumsetzungsfähigkeit erweitert worden. Somit stellt ArcInfo die komplette GIS in einem. Der ArcInfo-Desktop enthält alle Funktionen von ArcEditor und ergänzt diese um ein Set aus Datenverwaltung, Analyse, Konvertierungshilfsmittel und die ArcToolbox-Anwendung. Mit diesen Hilfsmitteln können Datenumsetzungen, Verallgemeinerungen, Anhäufungen, Testblätter, Bufferkreationen, statistische Berechnungen und vieles mehr durchgeführt werden.

Die ArcInfo-Workstation liefert hierzu noch das Geoprocessing über die klassischen Benutzerschnittstellen wie ARC, ArcEdit, ArcPlot, die ARC Makrosprache (AML) und mehr.

Zusätzlich zur Bereitstellung der bekannten Umgebung und vieler zahlloser bereits bestehenden GIS-Applikationen, stellt ArcInfo-Workstation fundamentale und unvergleichliche Geoprocessing-Funktionen zur Verfügung. ArcInfo-Workstations operieren auf Windows NT, Windows 2000 und UNIX-Plattformen.

4.1.3.1 ArcToolbox

ArcToolbox ist für ArcInfo ein komplettes Werkzeugset an Hilfsmitteln. Mehr als 170 Möglichkeiten für Geoprozesse, Datenumsetzungen, Manipulationen, Kartenseiteneinteilungen, Testblattanalysen und Karten Projektionen. ArcToolbox ist für ArcView und ArcEditor und enthält 6 geläufige benutzte Hilfsmittel für Datenumsetzung und Management. ArcToolbox wird von ArcInfo mitgeliefert jedoch nur eine vereinfachte Version mit ArcView und ArcEditor .

4.1.3.2 ArcCatalog

ArcCatalog erstellt und organisiert die geographischen und tabellarischen Daten. Er unterstützt bei der Erstellung von Metadata-Standards, der Bearbeitung und der Ansicht der Daten. Mit ArcCatalog kann man die GIS Datenausschnitte ansehen, geographische Informationen in einer Vorbetrachtung sehen, Tabellen direkt bearbeiten und er definiert die Schema-Struktur für die geographischen Datenschichten.

4.1.3.3 ArcMap

ArcMap ist die Desktop-Anwendung für alle kartenbasierenden Aufgaben einschließlich der Kartographie, der Kartenanalyse und der Bearbeitung. ArcMap gibt die Möglichkeit Daten zu visualisieren, zu erstellen, zu lösen, darzustellen und zu entwickeln.

disy Informationssysteme GmbH wurde 1997 in Karlsruhe gegründet. Sie war eine Ausgliederung des Forschungszentrums Informatik (FZI). Sie dient der langfristigen Pflege, Weiterentwicklung und Vermarktung von Software-Systemen. Durch den ständig wachsenden Kundenstamm wurde die Software für die Nutzer weiterentwickelt. Heute sind hierfür 22 Mitarbeiter angestellt. Im Mittelpunkt steht die Optimierung der Prozesse, indem die Daten eines Unternehmens zusammengeführt, strukturiert, vernetzt und präsentiert werden können. disy ist unabhängig von Systemen und Branchen und damit flexibel für den Nutzer. Grundlage der disy-Informationssysteme ist Java und XML und damit die neueste Intra- und Internettechnik.

4.2.1 GISterm

GISterm ist ein Geodatenzugriffssystem. Es ermöglicht auf räumlich verteilte Geodaten zuzugreifen und im Intra- oder Internet zur Verfügung zu stellen. Die verschiedenen Geodatenquellen können dann zu Karten zusammengestellt werden. Diese Karten können weiterverarbeitet werden zu thematischen Karten, sie können exportiert werden, in Office-Dokument übernommen, auf den Drucker ausgegeben oder als GIF-Bild gespeichert werden. Dies alles ist auch dem ungeübten GIS-Anwender möglich.

Beim Starten wird ein GIS-View automatisch geöffnet. Dieser View stellt eine Computerkarte dar, die mehrere Layer enthalten kann. Jeder Layer enthält unterschiedliche Karteninhalte. Das heißt, dass mehrere Kartenthemen vorstellbar, als „durchsichtige Folien“, übereinander gelegt eine Gesamtkarte ergeben. GISterm ist also ein Hilfsmittel, um kartographische Einheiten zu ordnen, zusammenzustellen und in einer Karte darzustellen. Diese Karten werden als AML-Dateien abgelegt und können so mit Hilfe von MapServer ins Intra- bzw. Internet gestellt werden. Dies wiederum stellt die Möglichkeit einer breiten Nutzung der Daten sicher.

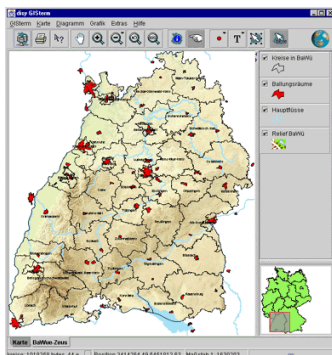


Abb. 6: GISterm

4.2.2 MapServer

Der MapServer ist eine Lösung, um interaktive Karten schnell und einfach anzubieten. Auch Laien ist es möglich, hiermit Geodaten bereitzustellen. Die Darstellung der Daten findet im Browser statt, die Kartenerzeugung auf dem Server. Die thematischen Karten werden mit GIS-Tools erzeugt und geladen. Dem Nutzer kann so eine stets aktuelle Darstellung der Geodaten gewährleistet werden. Bei der Anwendung des MapServers im Intra- oder Internet durch den Nutzer, kann dieser selbständig navigieren und zoomen und sich somit individuell mit den Daten vertraut machen.



Abb. 7: MapServer

4.3 VB (Visual Basic)

Visual Basic ist eine noch verhältnismäßig junge, objektorientierte Programmiersprache, die speziell für Windows entwickelt wurde. Mit dieser Sprache können schnell und einfach Programme für Windows programmiert werden. Sie verfügt über einen integrierten Accesszugriff, das heißt ein aus dem Datenbankprogramm Access übernommener Bestandteil unterstützt die Verwaltung von Datenbanken.

4.3.1 VBA (Visual Basic for Applications)

Aus dem Englischen übersetzt "Visual Basic für Anwendungen". Mit dieser Implementierung können Programm-Entwickler Dritten die Möglichkeit liefern, das jeweilige Programm um eigene Funktionen zu erweitern.

4.3.2 VB Script

Ist eine Untermenge des VisualBasic-Programmiersystems von Microsoft. VBScript-Programme können vom Microsoft Internet Explorer ab der Version 3.0 und auch von anderen Browsern gelesen werden. VBScript-Programme können entweder auf dem Computer, auf dem der Browser installiert ist oder auf dem World-Wide-Web-Server ausgeführt werden. VBScript hat einen geringeren Funktionsumfang als Visual Basic for Applications bzw. Visual Basic.

4.4 Der Douglas/Peucker-Algorithmus

Der Douglas-Peucker-Algorithmus ist eine Methode, um räumliche Ausreißer zu beseitigen. Mit dieser Option vereinfacht er die Linienzüge bei einem plötzlichen Richtungswechsel und löscht hierfür die überflüssigen Stützpunkte der Polygone. Der Algorithmus beginnt mit einer sehr groben Näherung und verfeinert diese dann. Er versucht nicht das Original zu vereinfachen. Er beginnt mit zwei Punkten p_1 und p_2 aus dem Polygon P , die am weitesten voneinander entfernt sind. Danach werden je nach Toleranz die Ausreißer gesucht. Hierfür wird die Näherung P' gebildet. Mit dem am weitesten entfernten Punkt wird begonnen. Dadurch wird die Verfremdung von P zu P' minimiert und die Form gelangt durch weitere Wiederholungen immer weiter zurück zum Original. Es werden immer weiter Punkte eingefügt, bis keine Originalpunkte mehr außerhalb der Toleranz liegen. Je kleiner also die Toleranz, je feiner die Wiederherstellung des Polygons. Die Datenmengen werden reduziert. Dieser Algorithmus hat ein Laufzeitverhalten von $O(nk)$, wobei k die Anzahl der hinzugefügten Punkte ist.

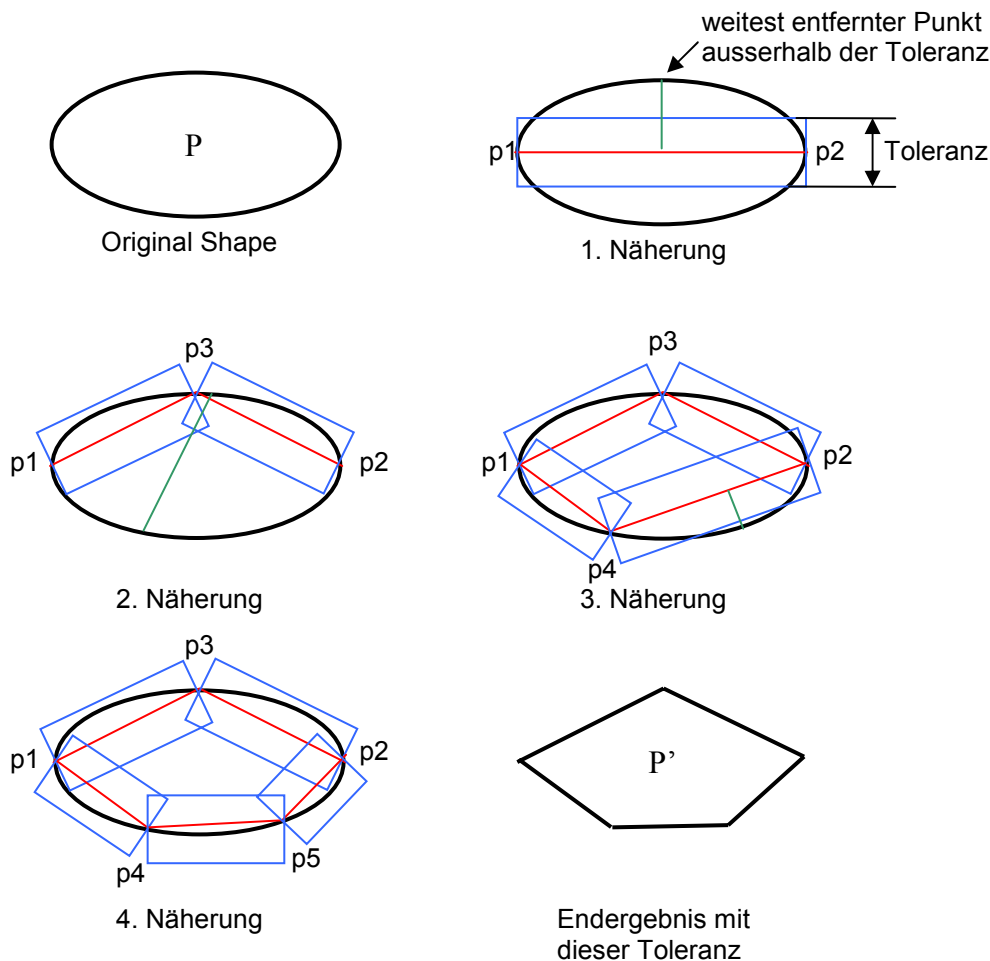


Abb. 8: Douglas/Peucker-Algorithmus

5 Bearbeitung der Geodaten

Für die Bearbeitung der Geodaten standen mehrere Möglichkeiten zur Verfügung. Es wurden verschiedene Methoden getestet. Im folgenden sind diese beschreiben, bis hin zur entgeltigen Version, dem Automaten.

5.1 Rastergeneralisierung

5.1.1 Rastergeneralisierung mit ArcView GIS

Um die Quelldaten der Naturschutzgebiete, der Biotope usw. in Rasterformat zu bringen, benötigt man bei diesem Verfahren das Programm ArcView GIS. Ist ArcView geöffnet, müssen die Originaldaten geladen werden. Sie werden mit Hilfe der Buttons und der dahinter verborgenen Schaltflächen geöffnet. Um sie zu bearbeiten werden sie durch „Maus-Klick“ aktiviert.

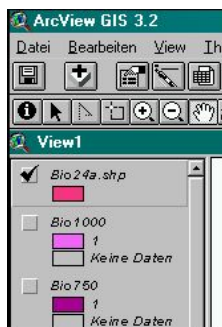


Abb. 9: Aktivierte Quelldaten

Der Befehl für die Umkonvertierung der Daten von Vektor in Rasterdaten, befindet sich in der Menüleiste unter „Thema“ -> „Convert to Grid ...“

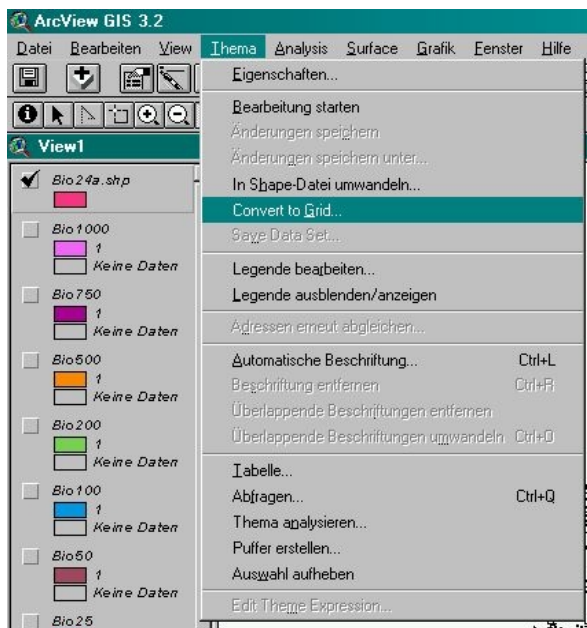


Abb. 10: Menüleiste mit Popup „Thema“

Bevor die Konvertierung beginnt, soll der Speicherort bekannt gegeben werden:

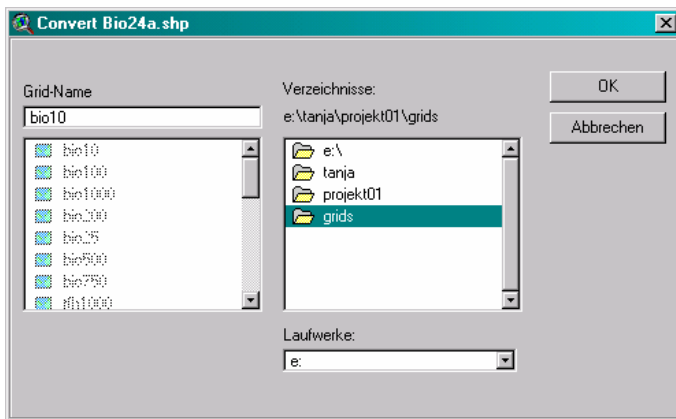


Abb. 11: Speichern unter ...

Nun müssen noch ein paar Angaben gemacht werden, welche die Generalisierung betreffen. So muss unter „Output Grid Extent“ der Name der Quelldaten stehen und die Größe der Pixel angegeben werden. Dies geschieht unter CellSize. Die Zahl entspricht der Meterangabe. Auf der folgenden Abbildung entspricht ein Pixel 10 m auf der Erdoberfläche. Die restlichen Felder bleiben unverändert. „Number of Rows“ und „Number of Columns“ werden vom Rechner selbst, bezüglich der Angabe in „CellSize“ ausgegeben.

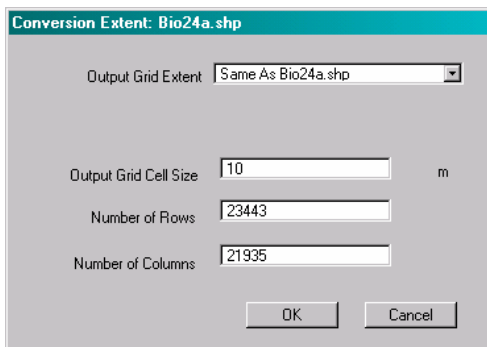


Abb. 12: Einstellungen

Um eine eventuelle Legende zu erstellen, benötigt der Rechner Informationen nach welcher Spalte diese zu erstellen ist. Bevorzugt wird eine Spalte, die in allen Zeilen identische Einträge hat, so dass nur zwei Legendeneinträge entstehen, nämlich „Daten“ und „keine Daten“ (sonst Effekt von Abb. 24). Das führt dazu, dass bei der Ausgabe nur eine Farbe vergeben wird und nicht für jeden Wert eine eigene.

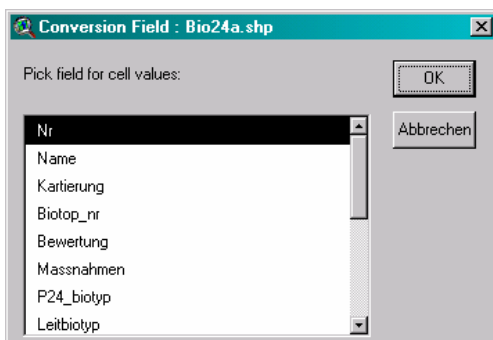


Abb. 13: Zu bearbeitendes Feld

Nachdem alle Angaben gemacht und mit OK bestätigt wurden, beginnt das System mit der Berechnung. Die benötigte Arbeitszeit hängt von der eingegebenen Pixelgröße ab, je kleiner je länger. In der Statusleiste wird zusätzlich eine Prozentangabe ausgegeben, wie weit die Berechnung fortgeschritten ist. Wenn die Berechnungen beendet sind, folgen zwei Abfragen, die erste betrifft die Verbindung der Programmattribute mit dem Grid. Um keinen unnötigen Speicherplatz zu verbrauchen, genügt hier die Eingabe „Nein“. Die zweite Abfrage betrifft die Ausgabe. Wenn man das Ergebnis gleich anschauen möchte, muss sie mit „ja“ bestätigt werden. Wurde dies getan, wird das Rasterbild im View geöffnet. Dieses befindet sich in der Legende an erster Stelle und muss mit einem Mausklick aktiviert werden, um es am Bildschirm sehen zu können.

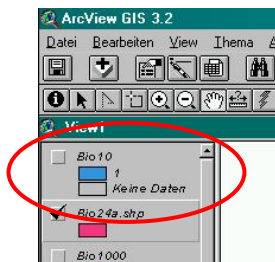


Abb. 14: Legende mit neuen Rasterdaten „Bio10“

Nun sind die Quelldaten gerastert. Um sie weiterhin nutzen zu können - Programme wie „GISterm“ verlangen das Dateiformat .shp (Shapes) - müssen die Daten ein weiteres Mal bearbeitet werden. Dazu war es dienlich ein Skript zu schreiben. Hiermit wird das aktuelle, aktive Thema in ein Shape gewandelt.

```

Raster in Shape umwandeln

'grid2shape.ave
'Verwandelt das aktive Thema in ein Polygon-Shapefile
'und fügt das Shapefile als ein Polygonthema in den View ein
'Voraussetzungen ist, dass nur ein GTheme aktiv ist!

'Benutzerabfrage
doGen = msgbox.YesNo ("Möchten Sie Polygone generalisieren?",
    "TEC7113 Grid to Shape Utility", false)

theView = av.GetActiveDoc
theTheme = theView.GetActiveThemes.Get(0)
theGrid = theTheme.GetGrid
'aFileName = "e:\tanja\projekt01\shape\lsgtest.shp".AsFileName

DefaultName=FileName.Make("e:\tanja\projekt01\shape").MakeTmp("shape", "shp")
theFileName=FileDialog.Put(DefaultName, "*.shp", "Shapedatei speichern")
if (theFileName=nil) then exit End

aPrj = theView.GetProjection
theResult = theGrid.AsPolygonFTab(theFileName, doGen, aPrj)
theFTheme = FTheme.Make(theResult)
theView.AddTheme(theFTheme)

```

Abb. 15: Skript „Rasterdaten in Shapes umwandeln“

Um das Skript ausführen zu können, müssen die zu bearbeitende Rasterdaten aktiviert werden. Wie auf der folgenden Abbildung zu sehen ist, wurde Bio10 in der Legende aktiviert. Folgend muss das Skript aktiviert werden, in dem man mit der Maus in das Skript oder auf dessen Rahmen „klickt“. Das Skript ist aktiv, wenn der Rahmen die Farbe wechselt und sich die Menüleiste, in die „Skript-Menüleiste“ verwandelt. Nun muss das Skript ausgeführt werden. Das geschieht mit der Taste „Ausführen“, dem „laufenden Männchen“ in der Menüleiste.

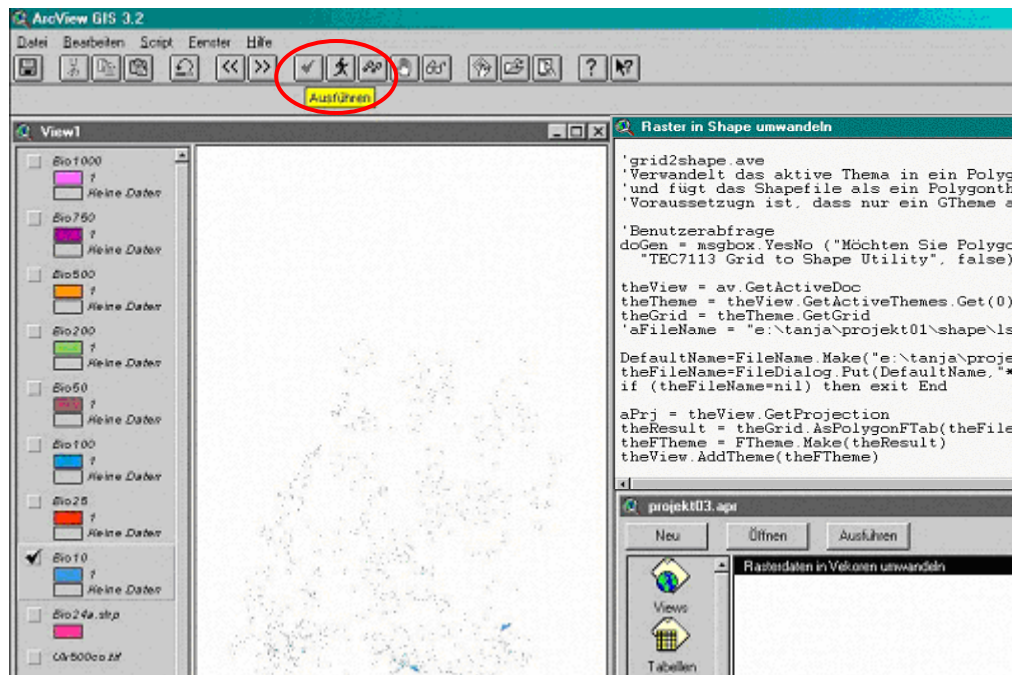


Abb. 16: Bildschirmübersicht

Das Script wird nun abgearbeitet. Es besteht die Möglichkeit, das Raster ein weiteres Mal zu generalisieren. Hierbei werden allerdings aus den Rasterquadraten Dreiecke, weshalb dies nicht zu empfehlen ist.

Danach muss der Name eingegeben werden, unter dem das Shape abgelegt wird. Der Pfad ist bereits vorgegeben.

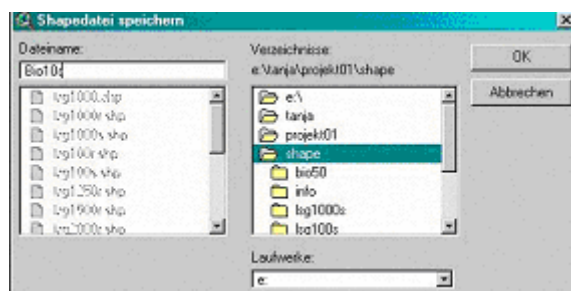


Abb. 17: Speichern

Ist dieser Vorgang beendet, werden die Daten in den View automatisch eingefügt. Um sie ansehen zu können, müssen sie aktiviert werden.

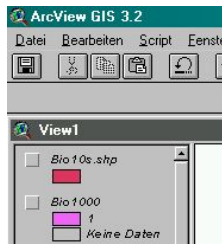


Abb. 18: Automatisch eingebunden

Im Vergleich kann man sehen, dass die Rasterdaten wieder zu einem Shape wurden mit dem vorher angegebenen Generalisierungsgrad. Erkennen kann man dies an der Darstellung von ArcView GIS, die shp-Daten werden mit einem Rahmen dargestellt. Darstellung und Farben können nun noch vom Benutzer manipuliert werden.

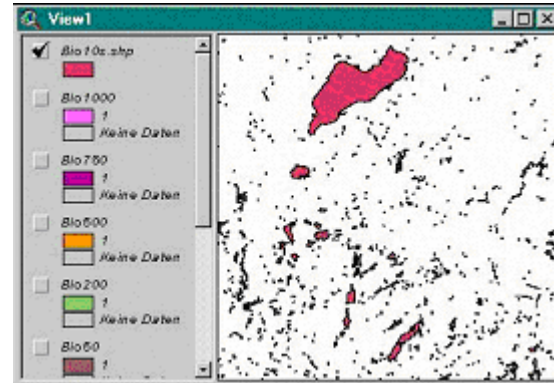
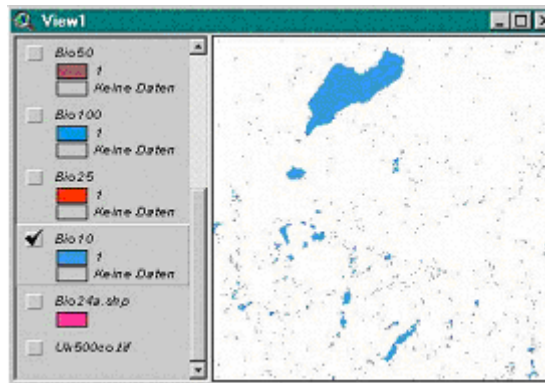


Abb. 19: Aus Raster wird Shape

5.1.2 Rastergeneralisierung mit ArcInfo

Zum Bearbeiten von Daten bietet ArcInfo den so genannten ArcCatalog an. ArcCatalog vereinfacht den Zugriff und das Handling von geographischen Daten. Als erstes werden die zu bearbeitenden geographischen Daten in den Catalog eingebunden. Man kann Ordner der lokalen Festplatte, der Datenbanken, aus dem Internet oder auch über ein erreichbares Netzwerk einbinden.

Nachdem der Katalog erstellt ist, kann nach den gewünschten Daten gesucht werden und die Resultate in verschiedene Sichten (Views) gezeigt werden. In ArcCatalog können alle Daten auf die selbe Art und Weise bearbeitet werden, unabhängig von ihrem Format unter dem sie gespeichert wurden.

Ist ArcCatalog geöffnet, müssen als erstes die Datenpfade verknüpft werden. Es muss der Ordner angegeben werden, der die zu generalisierenden Dateien enthält.

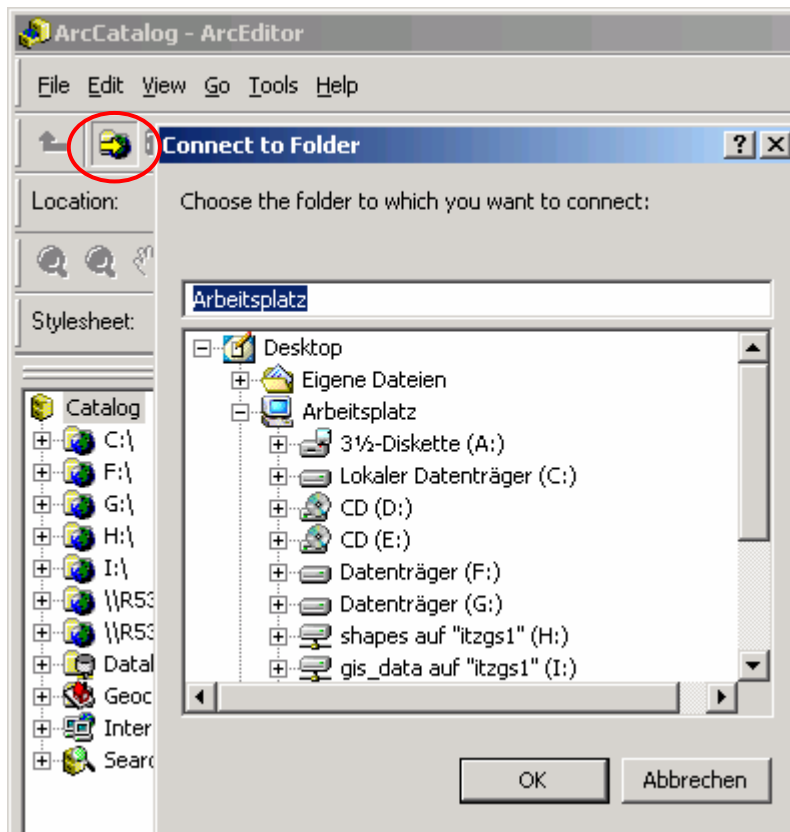


Abb. 20: „Verbindung“

Nun kann mit der Bearbeitung der Daten begonnen werden. ArcInfo stellt zur schnelleren Bearbeitung einen „Werkzeugkasten“ namens ArcToolbox zur Verfügung, der die möglichen Befehle beinhaltet. ArcToolbox liefert ein reiches Angebot an über 170 Werkzeugen, die dem Nutzer zur Verfügung stehen. Es ist ohne Schwierigkeiten möglich das Tool zu finden, das man sucht. Mit ArcToolbox können sichtbar alle großen Gebietsdatensätze in und aus ArcInfo konvertiert werden.

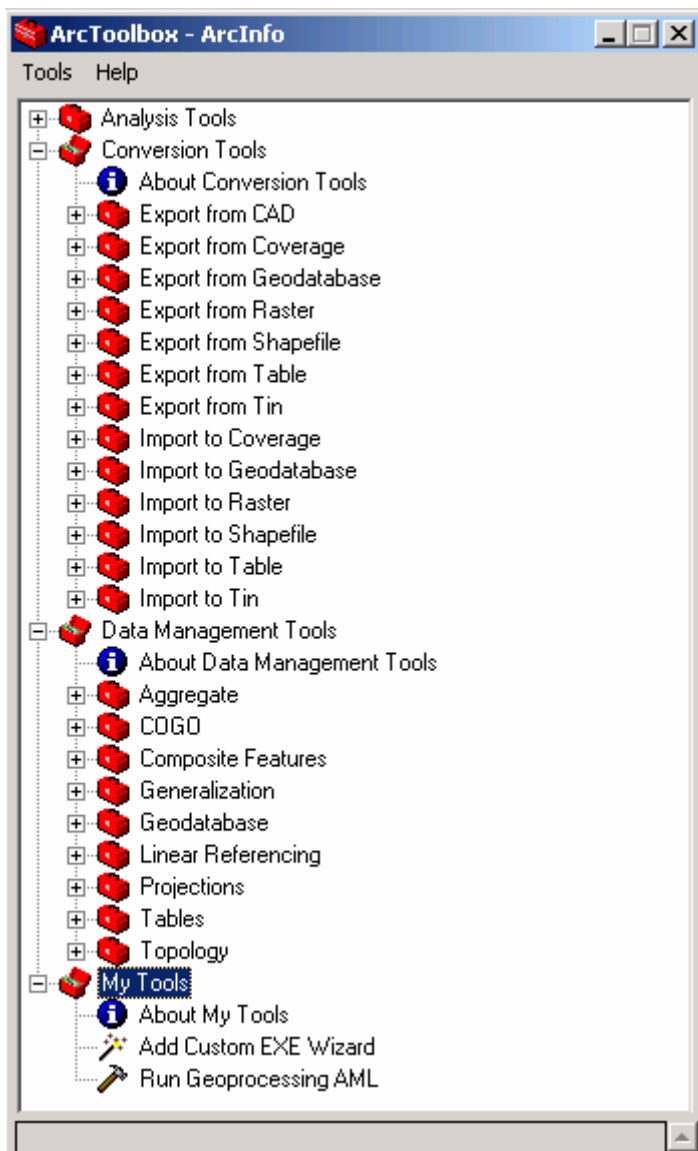


Abb. 21: ArcToolbox

Zuerst müssen die Daten in ein Coverage umgewandelt werden. Hierfür stellt ArcToolbox unter „Konvertierungs-Werkzeuge“ -> „Import zu Coverage“ -> „Shapefile zu Coverage“ eine Schaltfläche zur Verfügung (s. folgende Abbildung). Es öffnet sich diese Benutzerabfrage in die folgendes einzugeben ist, um die gewünschten Effekte zu erreichen: unter „Eingabe-Shapefile“ muss das zu bearbeitende Shape aufgerufen werden, dazu steht ein Assistent zur Hilfe, der die Suche nach den Daten vereinfacht. Der „Ausgabe-Feature-Class“ muss „Poly“ beinhalten und unter „Ausgabe-Coverage“ muss die Datei angegeben werden, unter der man das Coverage ablegen will. Mit einem Klick auf OK beginnt die Umwandlung.

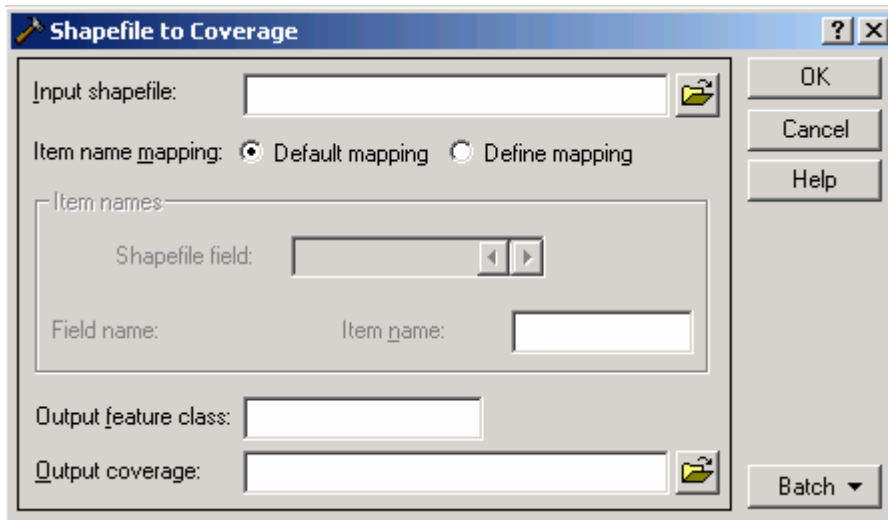


Abb. 22: vom Shapefile zu Coverage

Um das Coverage zu einem Grid umzuwandeln und gleichzeitig zu generalisieren, stellt ArcToolbox unter „Konvertierungs-Werkzeuge“ -> „Import zu Raster“ -> „Polygon-Coverage zu Grid“ eine weitere Schaltfläche zur Verfügung (s.u.). Wieder öffnet sich eine Benutzerabfrage in die folgendes einzugeben ist:

unter „Eingabe-Coverage“ muss nun die in eine Coverage gewandelte Shapedatei stehen, die zu einem Grid verarbeitet werden soll. Es stehen mehrere Konvertierungsoptionen zur Auswahl. Als erstes kann man unter „Werte-Feld“ die Spalte angeben, nach der sortiert werden soll. Es empfiehlt sich hier eine Spalte auszuwählen, die einen einheitlichen Wert enthält, da sonst jedem Wert eine andere Farbe zugewiesen wird.

Des Weiteren können Gewichtung- und Lookup-Tabellen angegeben werden.

Unter „Zellengröße“ kann nun die Stärke der Generalisierung angegeben werden.

Wenn der Wert 100 eingegeben wird, entspricht dies einer Vermaschung von 100 m auf der Erdoberfläche (1:1). Darunter gibt der Rechner automatisch die Anzahl der Zeilen und Spalten, die sich dadurch ergeben aus.

Nun muss nur noch der Pfad und ein Name für die Speicherung angegeben werden.

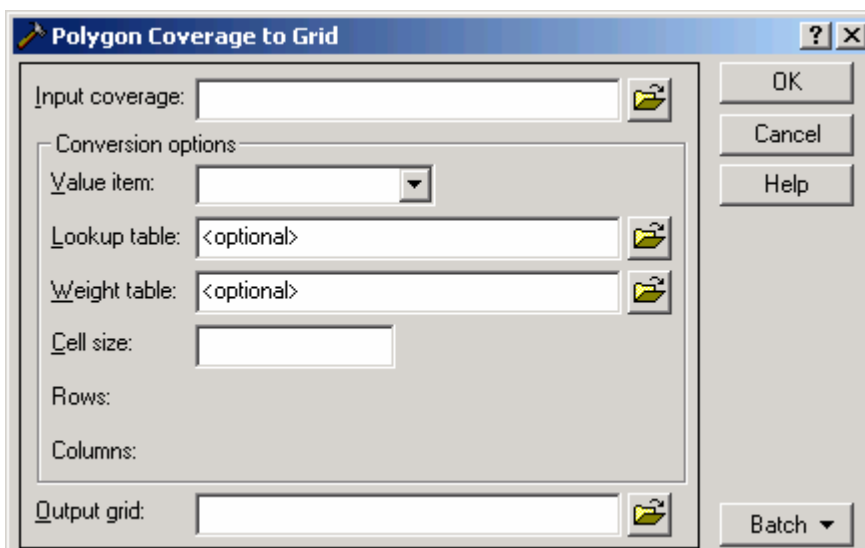


Abb. 23: Polygon-Coverage zum Grid

In der Vorschau mit der Angabe der Geographie kann man das Ergebnis anschauen (s.u.). Bei diesem Beispiel, war die Spalte nach der sortiert wurde (Isg25Cover#) jeweils mit einem anderen Wert gefüllt, so dass der bereits erwähnte Effekt eintrat und jedes der Felder einen anderen Grauwert bekam.

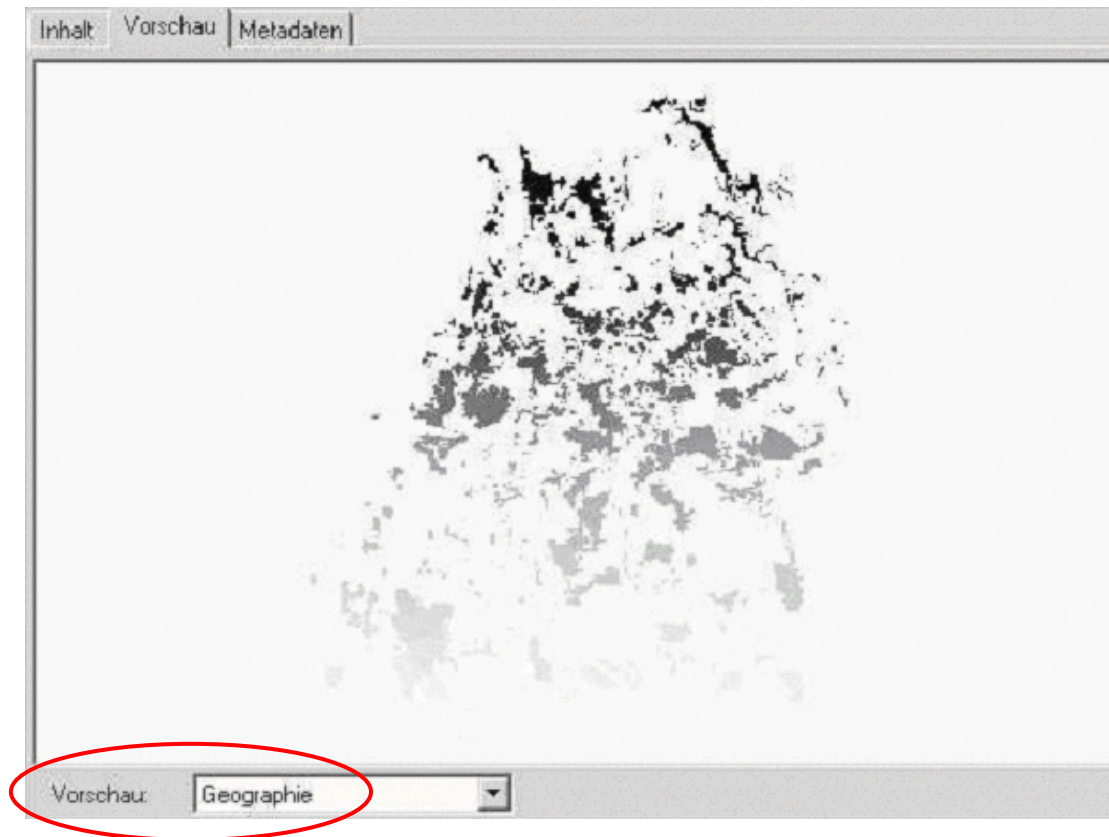


Abb. 24: Farbverlauf

Um diesem Effekt zu umgehen, muss man eine Spalte auswählen, die in allen Zeilen den selben Wert enthält. Notfalls muß hierfür eine Spalte definiert werden. Dies wurde bei dem folgenden Beispiel mit der Spalte namens „eins“ umgesetzt.

	NR_ALT	FFC	OAC	GEOM_ID	OBJECT_ID	EINS
▶	1.05.002	10	13	1	81280000002	1
	1.05.002	10	13	1	81280000002	1
		0	0	0		1
	1.05.005	10	13	1	81280000005	1
	1.05.002	10	13	1	81280000002	1
		0	0	0		1
		0	0	0		1
	1.05.002	10	13	1	81280000002	1
	1.05.005	10	13	1	81280000005	1
		0	0	0		1
	1.05.002	10	13	1	81280000002	1
	1.05.010	10	13	1	81280000010	1
	1.05.002	10	13	1	81280000002	1
	1.05.002	10	13	1	81280000002	1
	1.05.010	10	13	1	81280000010	1
		0	0	0		1
		0	0	0		1

Datensatz: 1 Zeigen: Alle Ausgewählte Datensätze: (von 3877)

Vorschau: **Tabelle**

Abb. 25: Lösung

Die Abbildung zeigt die Graphik einfarbig nach der Bearbeitung der veränderten Datei.

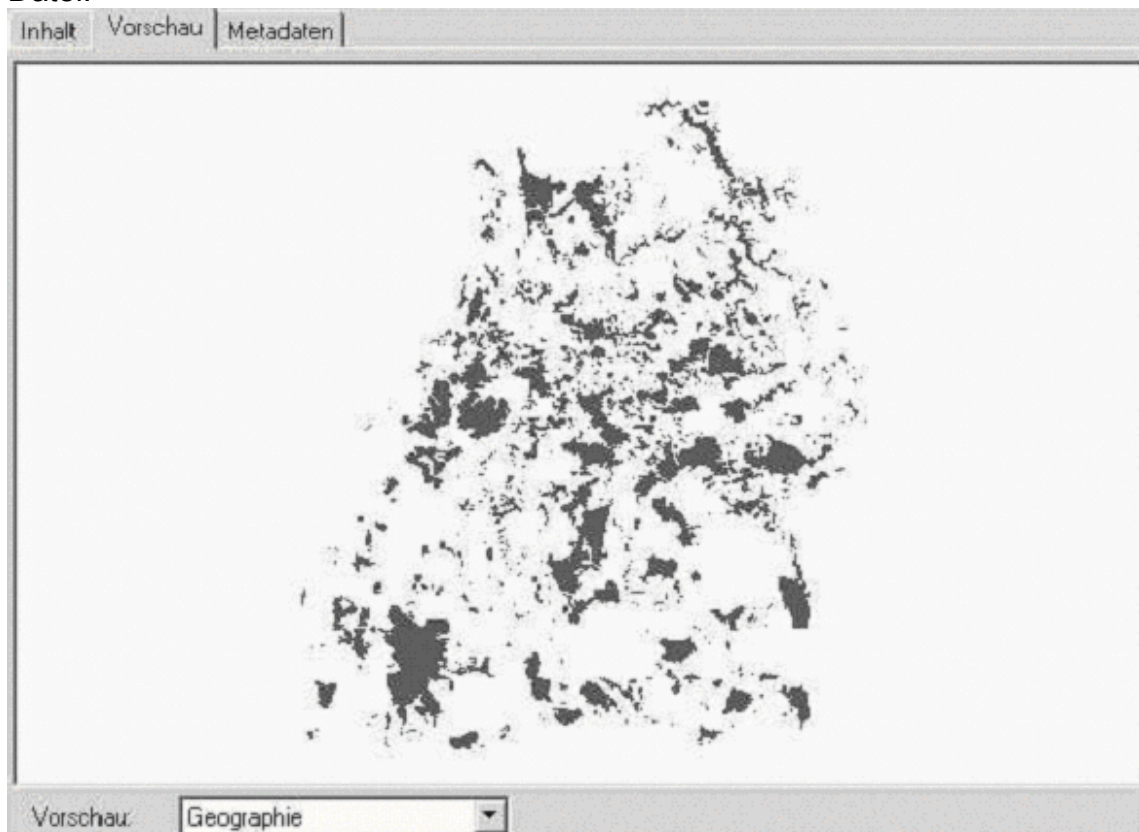


Abb. 26: Vorschau einfarbig

5.2 Vektorgeneralisierung

5.2.1 Vektorgeneralisierung mit ArcGIS

Um die Daten in ArcInfo bearbeiten zu können, müssen sie zuerst vom Server geladen werden. Hierfür steht das Programm LeechFTP hilfreich zur Seite. Die benötigten Daten die man vom Server laden möchte, müssen hier nur markiert werden und mit dem Button „Upload Files“ bestätigt werden. Die geladenen Daten befinden sich nun auf dem gewünschten Laufwerk und man kann mit der Bearbeitung beginnen.

Nun wird der Xmanager geöffnet. Über diesen gelangt man auf eine Unix-Oberfläche, auf der ArcInfo-Workstation liegt.

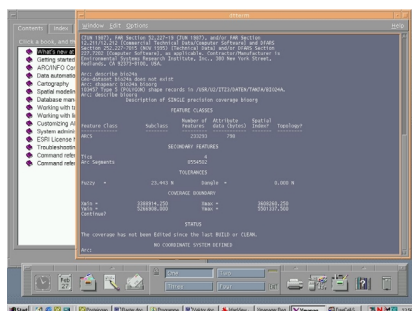


Abb. 27: Unix-Oberfläche

Mit dem Befehl ARC wird das ARC-System initialisiert und der Prompt erscheint als *Arc:*.

Folgende Befehlsreihenfolge war für die Bearbeitung der Biotope nötig:

```
Arc: shapearc bio24a bioorg bio  
Arc: list bioorg.patbio  
Arc: build bioorg poly  
Arc: clean bioorg bioorg  
Arc: generalize bioorg bio1000v 1000 bendsimplify  
Arc: build bio1000v poly  
Arc: clean bio1000v bio1000v  
Arc: arcshape bio1000v regions.bio bio1000v
```

Befehlserklärungen:

- **shapearc bio24a bioorg bio**
es werden aus Shapes Linien gebildet. (Befehl, Quelle-, Zieldatei, Unterklasse)

Syntax:

```
shapearc <in_shape_file><out_shape_file>{out_subclass}{default|define}
```

- **list bioorg.patbio**

zeigt die Records der Datei

Syntax:

```
list <info_file>{from}{to}{item...item}
```


- **build bioorg poly**

erstellt oder erneuert die Eigenschaften der Attributtabelle für eine Coverage.

Syntax:

build <cover>{poly|line|point|node|anno.<subclasses>}

- **clean bioorg bioorg**

erstellt eine Coverage mit korrekten Polygonen oder korrekter Knotenpunkttopologie. Korrigiert „Errors“ in der Geometrie.

Syntax:

clean <in_cover>{out_cover}{dangle_lenght}{fuzzy_tolerance}{poly|line}

Die Daten sind nun vorbereitet und können generalisiert werden. Hierzu müssen folgende Befehle in ArcGIS abgearbeitet werden:

- **generalize bioorg bio1000v 1000 bendsimplify**

Generalisierungsbefehl (Befehl, Quell-, Zieldatei, Faktor, Art der Generalisierung)

Syntax:

generalize <in_cover><out_cover><weed_tolerance>{pointremove|bendsimplify}

- **build bio1000v poly** erstellt oder erneuert wiederum die Eigenschaften der Attributtabelle für eine Coverage.

Syntax: siehe build-Befehl oben

- **clean bio1000v bio1000v**

erstellt eine Coverage mit korrekten Polygonen oder korrekter Knotenpunkttopologie. Korrigiert „Errors“ in der Geometrie.

Syntax: siehe clean-Befehl oben

- **arcshape bio1000v regions.bio bio1000v**

schreibt Attributeigenschaften oder INFO-Records in eine neue Shapefile.

Syntax:

arcshape <in_cover><in_feature_class><out_shapefile>{default|define}

Die Daten sind nun generalisiert und aufbereitet. Sie müssen wieder zurück auf den Server geladen werden, damit sie in allen anderen Programmen aufgerufen werden können.

Um sie mit GISterm aufrufen zu können, müssen sie nochmals in ArcView geladen werden. In der Thementabelle müssen alle Spalten, die mit einem „_“ neu angelegt wurden, gelöscht werden. Da dies zu einem Lesefehler von GISterm führt, der nicht anders behoben werden kann. Dies ist nicht wünschenswert, hier für muss noch eine andere Lösung gefunden werden.

Nun können die generalisierten Biotopdaten in allen Graphikprogrammen, die shp-Dateien verwalten können, aufgerufen und genutzt werden.

5.2.2 Vektorgeneralisierung mit ArcInfo

Um mit den Vektordaten in ArcCatalog arbeiten zu können, müssen diese zuerst, wie auch bei den Rasterdaten in Coverages umgewandelt werden. Ein Werkzeug hierfür findet man in der ArcToolbox unter „Konvertierungs-Werkzeuge“ -> „Shapefile zu Coverage“. Es erscheint eine Benutzerabfrage in die folgendes einzugeben ist: In das Feld „Eingabe-Shapefile“ muss der Pfad zu den Originaldaten eingegeben werden, unter „Ausgabe-Feature-Class“ muss Line stehen und unter „Ausgabe-Coverage“ kann ein neuer Name für die Coverage angegeben werden.

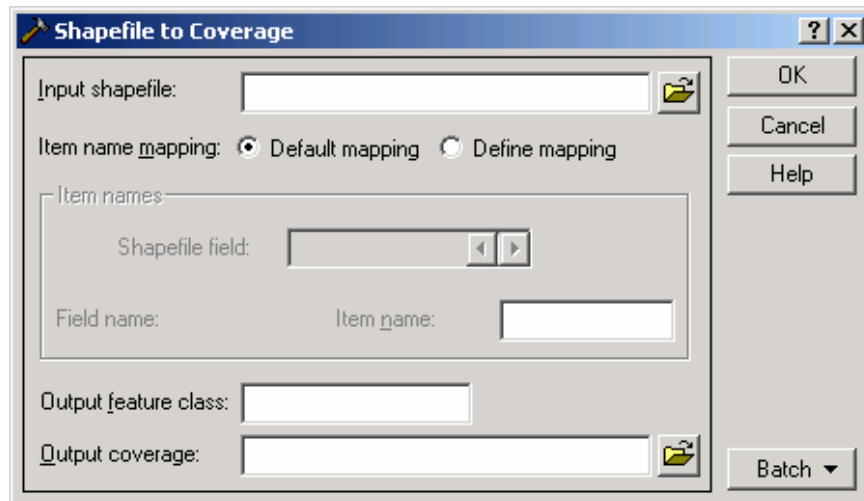


Abb. 28: Shape in Coverage umwandeln.

Zum Generalisieren der Daten, stellt ArcToolbox ein weiteres Werkzeug zur Verfügung, den man unter den „Datenmanagement-Werkzeugen“ -> „Generalisierung“ -> „Linien vereinfachen“ findet. Was passiert? Mit dem Werkzeug „Linien vereinfachen“ werden die Polygrenzen und diverse Details reduziert. Der Faktor der Reduzierung hängt von der angegebenen Toleranz unter „Ausdünnungs-Toleranz“ und der „Vereinfachung“, ob Punkte entfernt oder Kurven vereinfacht werden sollen, ab. Bei der Option „Punkte entfernen“ wird der Douglas/Peucker-Algorithmus angewandt, der an anderer Stelle beschrieben wird. Bei der Option „Kurven vereinfachen“ wird Formalysetechnik angewandt. Diese erkennt Biegungen. Ist eine Biegung zu eng, wird sie der Toleranz entsprechend geweitet und unwichtige Verläufe werden eliminiert. Die Qualität bei der entstehenden Linie ist besser und gleicht dem Original eher als beim Douglas/Peucker-Algorithmus.

Folgende Eingaben sind in die Benutzerabfrage einzugeben:

Unter „Eing.-Coverage“ muss das zuvor erstellte Coverage aufgerufen werden. Die „Ausdünnungs-Toleranz“ gibt die Stärke der Generalisierung in Metern an. Es können beliebige Werte eingegeben werden. Um die LSGs zu generalisieren bieten sich Werte zwischen 50 und 1000 an. Je höher die Zahl, je stärker die Generalisierung. Unter „Vereinfachung“ empfiehlt sich „Kurven vereinf.“ anzuklicken, da hier bestehende Kurven berücksichtigt werden. Unter „Punkte entfernen“ kommt es zu Formveränderungen, die nicht mehr dem Original zugeordnet werden können. Unter „Ausg.-Coverage“ kann wieder ein neuer Name für das generalisierte Objekt vergeben werden, sonst wird das alte überschrieben und das originale Shapefile würde nicht mehr existieren.

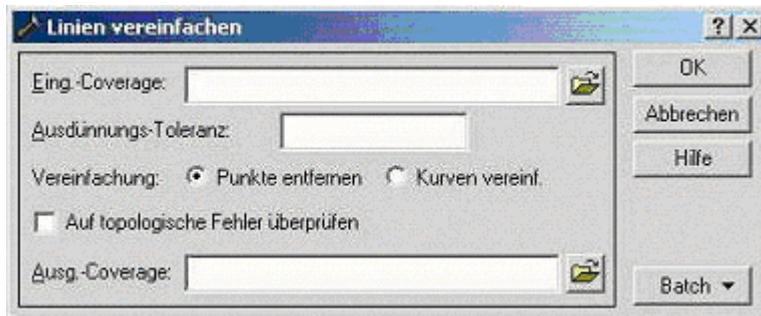


Abb. 29: Generalisierung.

Da die nun generalisierte Coverage Fehlerdaten enthalten kann, wird im nächsten Schritt eine Säuberung vorgenommen.

Unter „Topologie“ -> „Topologie erzeugen und Fehlerbereinigung“, was dem „clean“-Befehl in ArcGIS entspricht, erscheint ein weiteres Werkzeug aus der ArcToolbox. Folgende Eingaben wurden gemacht:

Unter „Eing.-Coverage“ muss die gerade generalisierte Datei stehen. „Dangle-Länge“ und „Fuzzy-Toleranz“ können belassen werden, es sei denn, man möchte den Abstand der Stützpunkte nochmals manipulieren. Die „Feature Class“ muss „Poly“ sein und unter „Ausg.-Coverage“ kann wieder ein neuer Name eingegeben werden. Die Wirkung vom „clean“-Befehl wird durch eine geometrische Analyse der Arcs und Lablepunkte angeführt, die die Nodes und Polygone identifiziert. Es werden die Polygon- und Arc-Node-Topologien erstellt. Die Kreuzungs-, Anfangs- und Endpunkte der Arcs werden als Nodes abgespeichert. Eventuell werden hierfür die Arcs gespalten. Der „clean“-Befehl erstellt eine neue Liste mit Arcs und Nodes, in der diese bereinigt und neu definiert werden. Dabei werden die Nodes neu nummeriert, so wie die benachbarten Flächen erfasst.

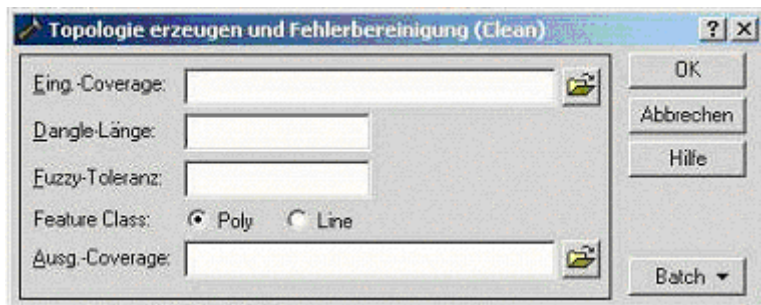


Abb. 30: Clean.

Zum Schluss muss für die Weiterverarbeitung in GIS-Tools die bereinigte Coverage zurück in ein Shapefile gewandelt werden. Hierfür benutzt man ein weiteres Werkzeug aus ArcToolbox. Unter „Konvertierungswerkzeuge“ -> „Export aus Coverage“ -> „Coverage zu Shapefile“ öffnet sich die Benutzerabfrage:

In die „Eingabe-Coverage“ kommt der Pfad der bereinigten Coverage, in der „Feature Class“ muss „Poly“ stehen und unter „Ausgabe-Coverage“ kann wieder ein neuer Name angegeben werden.

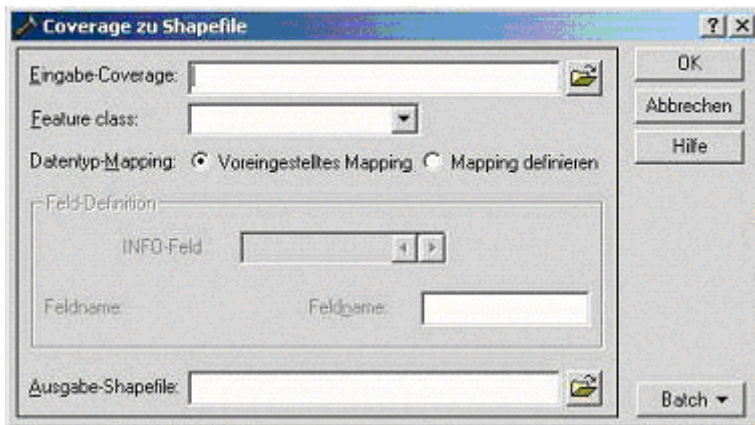


Abb. 31: Coverage zu Shapefile.

ArcToolbox stellt eine weitere Möglichkeit, um das Auffinden öfter genutzter Werkzeuge zu erleichtern, die einzelnen Werkzeuge werden unter „Meine Werkzeuge“ abgelegt. Dafür klickt man mit der rechten Maustaste auf das gewünschte Werkzeug, das in die eigene Werkzeugkiste soll, geht auf „senden an“ -> „Meine Werkzeuge“. Damit befindet sich ein Pfad zu dieser Funktion bei „Meine Werkzeuge“ unten in der Hauptwerkzeuggeste.

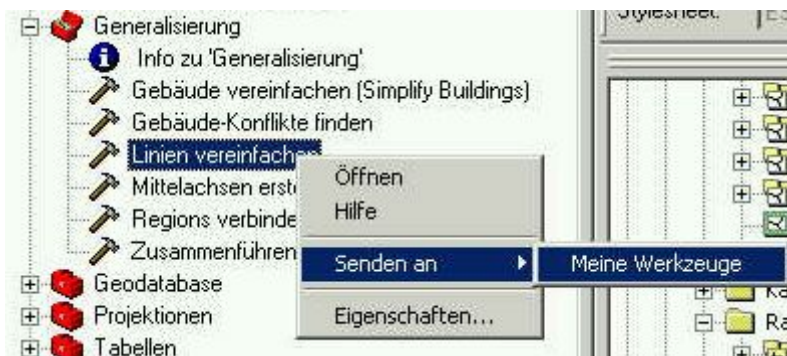


Abb. 32: Werkzeuge verwalten.

Hier kann nun noch ein eigener Name vergeben werden. Befehlsabläufe können zur Unterstützung nummeriert werden.



Abb. 33: „Meine Werkzeuge“ benennen.

5.3 Der Generalisierungsautomat

Die bis jetzt beschriebenen Generalisierungsmethoden sind mit viel Aufwand verbunden und stehen in keinem Verhältnis mit dem Nutzen. Der Automat hat die Aufgabe, die einzelnen Umweltdaten übergreifend für alle Massstäbe nutzbar zu machen ohne mehrere Befehle hintereinander benutzen zu müssen. Hierfür wurde die Programmiersprache VisualBasic (VB) erlernt, mit der Macros geschrieben werden können, die unabhängig von einem Programm genutzt werden können.

5.3.1 Erstellung der Benutzeroberfläche

Für den Benutzer muss eine einfach bedienbare und handhabbare Oberfläche geschaffen werden. Mit Hilfe der gegebenen Toolbox werden die einzelnen Elemente eingesetzt. Die zu diesem Zeitpunkt noch sichtbaren Orientierungspunkte dienen der besseren graphischen Aufteilung, des User-Forms und sind bei späterem Aufruf nicht mehr zu erkennen. Man klickt auf das gewünschte Werkzeug in der Toolbox und platziert es in der User-Form. Anschließend können noch Größe, Form oder Textinhalte verändert werden. Hier wurden zur Realisierung der Benutzeroberfläche mit Label, Textbox, Optionbutton und Commandbutton gearbeitet.

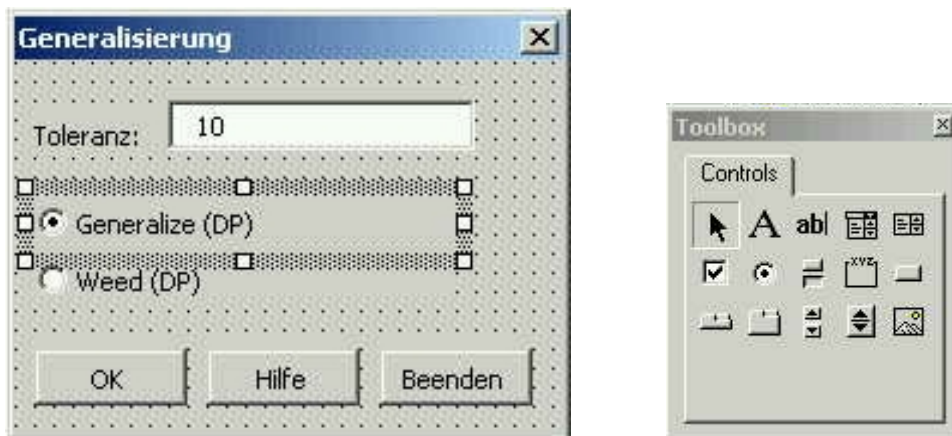


Abb. 34: Erstellung der Benutzeroberfläche und Toolbox

Um den einzelnen Buttons Funktionen zuzuweisen, müssen diese in Verbindung mit dem Makro gebracht werden. In der Oberfläche „Properties“ muss der Name des Options-Button angegeben werden. Als Beispiel hier der Options-Button „optGen“. Darunter befinden sich nun unter „Alphabetic“ die für diesen Button individuellen Einstellungen, wie z.B. der Name, die Größe und spezielle Effekte. Diese weisen auf das VB_Skript hin, in dem unter dem Namen des Buttons (hier optGen), die weiteren Optionen des Buttons festgelegt werden. Hier wird sichergestellt, dass nur eine der beiden Generalisierungsmöglichkeiten genutzt werden soll, also die „generalize“-Option und nicht die „weed“-Option.

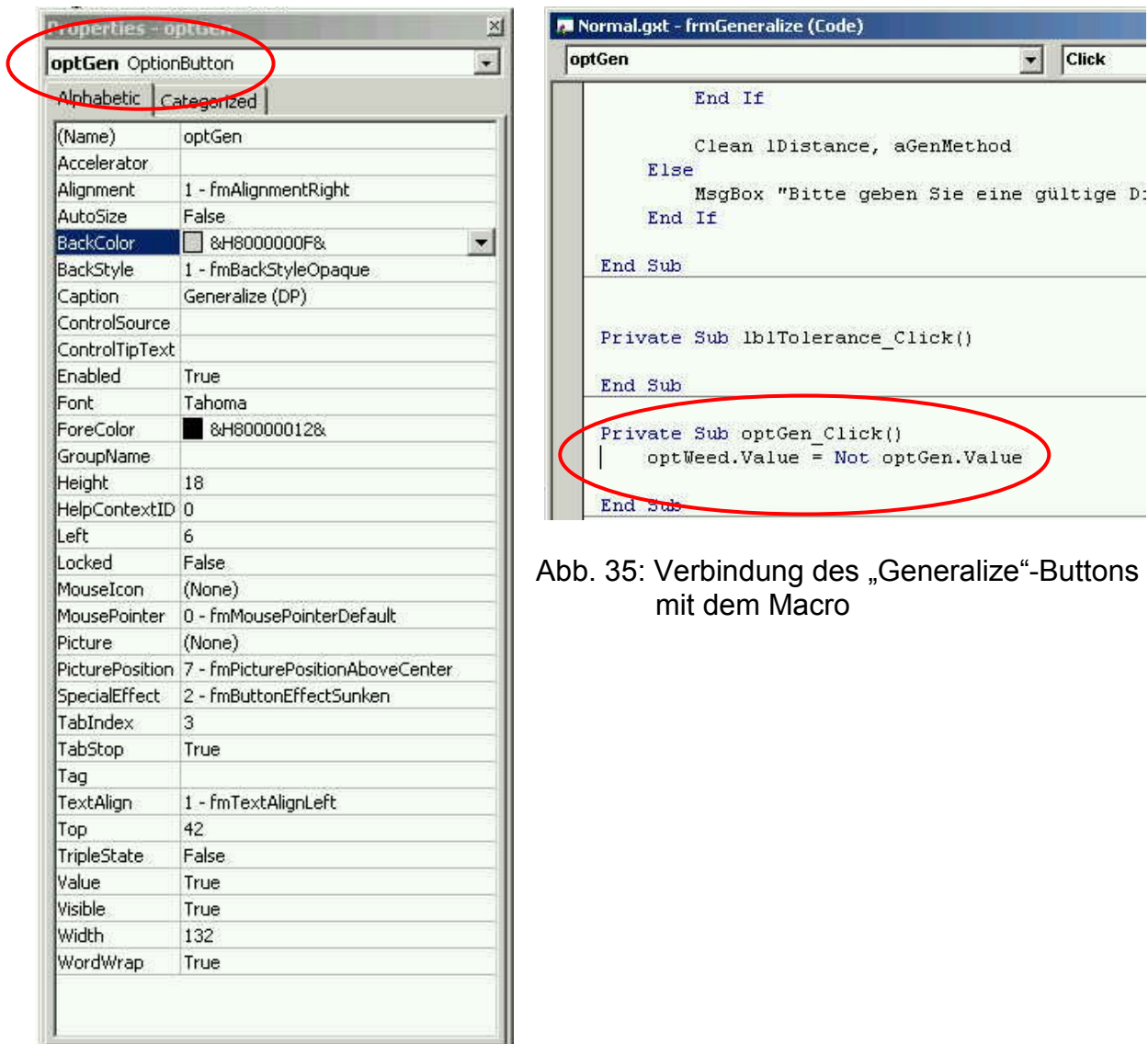


Abb. 35: Verbindung des „Generalize“-Buttons mit dem Macro

5.3.2 Erläuterungen zum Macro

Hintergrund der Schaltfläche in VB-Sprache:

[...]

```
Private Sub cmdGeneralize_Click()
```

```
    Dim lDistance As Long
```

```
    Dim aGenMethod As genMethod
```

} Festlegen der Variablen

```
    If IsNumeric(txtTolerance.Text) Then
```

```
        lDistance = CLng(txtTolerance.Text)
```

} Verarbeitung der Toleranz

```
        If optGen.Value Then
```

```
            aGenMethod = genGeneralize
```

```
        Else
```

```
            aGenMethod = genWeed
```

```
        End If
```

} Abfrage: Ist der Optionsbutton
generalize oder weed aktiv?
Zuweisung der jeweiligen
Methode.

```
        Clean lDistance, aGenMethod
```

} Bereinigung

```
    Else
```

```
        MsgBox "Bitte geben Sie eine gültige Distanz an!", vbInformation, "Achtung"
```

```
    End If
```

} Ausgabe bei
fehlerhafter
Eingabe

```
End Sub
```

[...]

Das komplette Macro befindet sich im Anhang.

5.3.3 Anwendung des Automaten

Die Generalisierung fand in der Programmiersprache VB (Visual Basic) statt. Mit dem folgenden Button wird das Macro gestartet. Diesen habe ich selbständig erzeugt, damit man die Arbeit mit einem einzelnen Klick beginnen kann, ohne Hintergrundwissen zu benötigen.



Abb. 36: Startbutton

Nach dem Aufruf des Macros mit Hilfe des Startbuttons oder direkt in der Macroebene, erscheint folgender Bildschirm. Hier müssen Angaben gemacht werden, über die Art und Stärke der Generalisierung. In dem Textfeld „Toleranz“ muss die Stärke der Generalisierung eingegeben werden. Zu empfehlen sind Werte zwischen 50 und 1000 für die Generalize-Methode und Werte zwischen 50.000 und 1 Mio. für die Weed-Methode. Welche Methode man benutzen möchte, ist darunter anzugeben. Sind alle vorbereitenden Schritte eingegeben, so kann die Generalisierung mit dem Button „OK“ begonnen werden. Werden hier keine Eingaben gemacht, wird für die Distance 10 und für die Methode Generalize benutzt.



Abb. 37: Benutzeroberfläche

Nun folgt eine Abfrage der Originaldaten, die generalisiert werden sollen. Die Daten können von allen zugänglichen Ordnern aus aufgerufen werden.

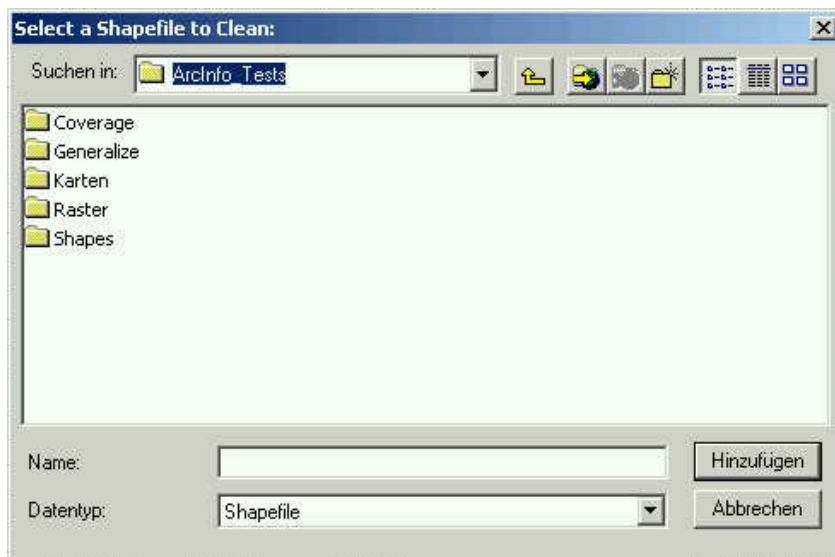


Abb. 38: Originaldaten

Wird kein Quell-Shapefile ausgewählt, so erhält man eine Fehlermeldung.



Abb. 39: Error

Anschließend ist eine Angabe zu machen, wo man die generalisierten Daten speichern möchte

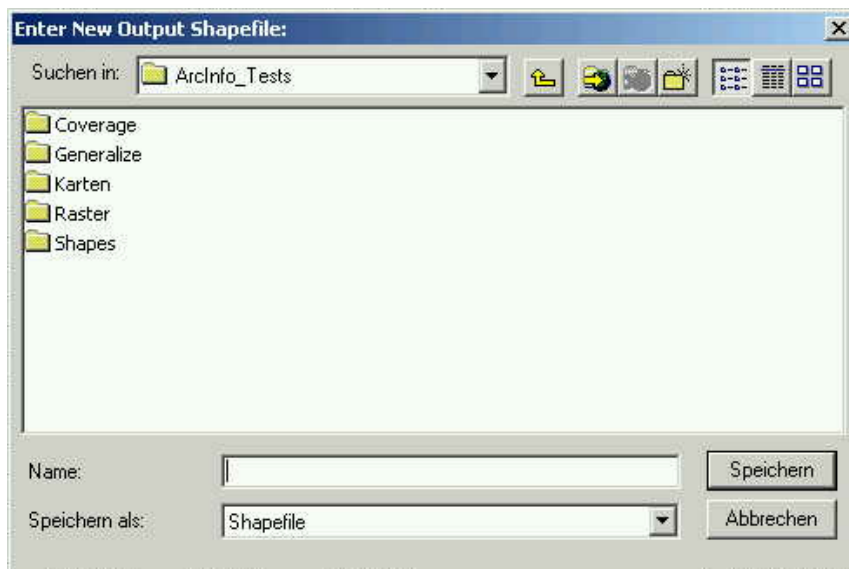


Abb. 40: Speichern unter ...

Nach dem alle Angaben vollständig sind, beginnt der Automat mit den Berechnungen. Währenddessen erscheint auf dem Bildschirm eine Prozessüberwachung, die Auskunft darüber gibt, wie viele Daten insgesamt vorhanden und wie viele bereits bearbeitet sind.



Abb. 41: Prozessüberwachung

Zum Abschluss wird die Bearbeitung bestätigt. Hier würden auch eventuelle Fehler während der Generalisierung oder nicht gefundene oder geschlossene Shapes gemeldet werden. In diesem Fall wurden alle 1505 Features korrekt bearbeitet und unter dem gewünschten File abgelegt.



Abb. 42: Abschlussangaben

5.3.4 Hilfe

Da der Automat individuell unabhängig von einem Programm Shapefiles umwandeln kann und im Intranet für Jedermann zur Verfügung gestellt wird, wurde hier eine kurze Einweisung gegeben. Dies war nötig, da die Benutzeroberfläche nicht selbsterklärend ist und für einen Laien nicht erkenntlich ist, mit welchen Werten er arbeiten kann. Es wird kurz beschrieben, worum es sich handelt und wie der Automat zu handhaben ist.

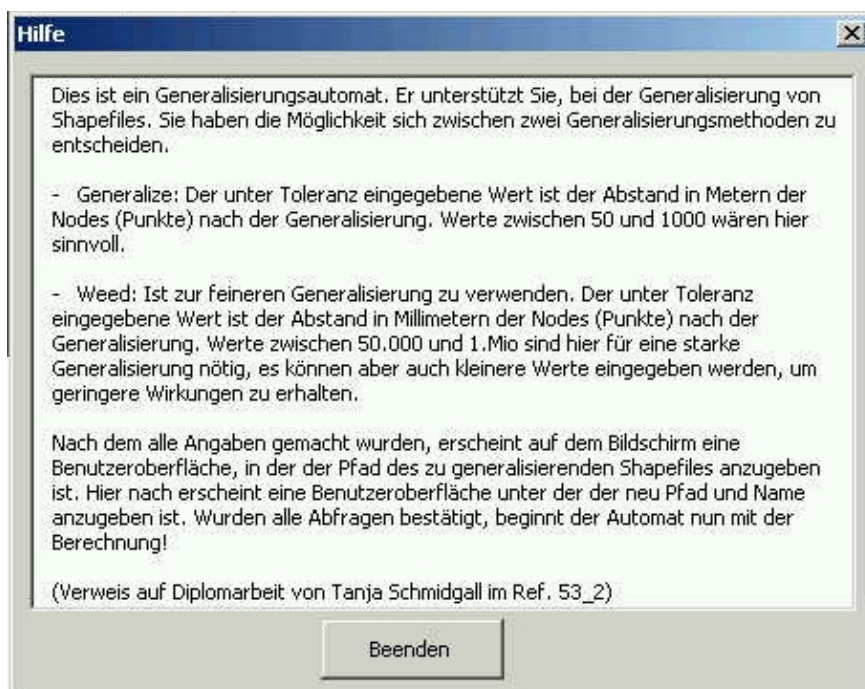


Abb. 43: Hilfefenster

5.3.5 Die Extension

Um den Generalisierungsautomaten unabhängig von einem Programm nutzen zu können, muss das in VB erstellte Macro mit allen Benutzeroberflächen und deren Quellcodes zu einem Projekt zusammengefasst werden. Diese Projekt bekam den Namen GeneralizeCommand.dll. Um diese an einem beliebigen Rechner ausführen zu können, muss das Projekt zuerst auf dem Rechner registriert werden. Dafür findet man unter „Start“ ->“Ausführen...“ die folgende Benutzeroberfläche. Hier muss die Registrierung und der Pfad angegeben werden.

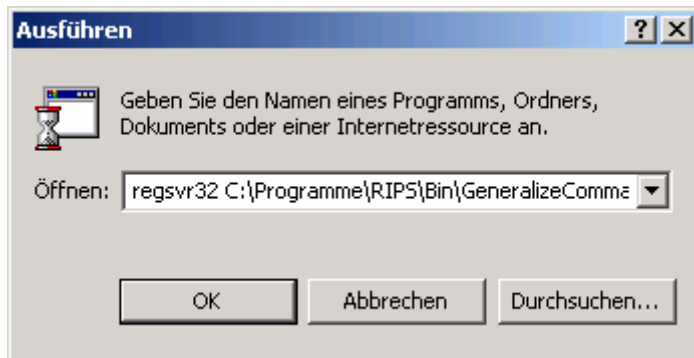


Abb. 44: Registrieren

Es ist auch möglich das Projekt direkt aufzurufen:

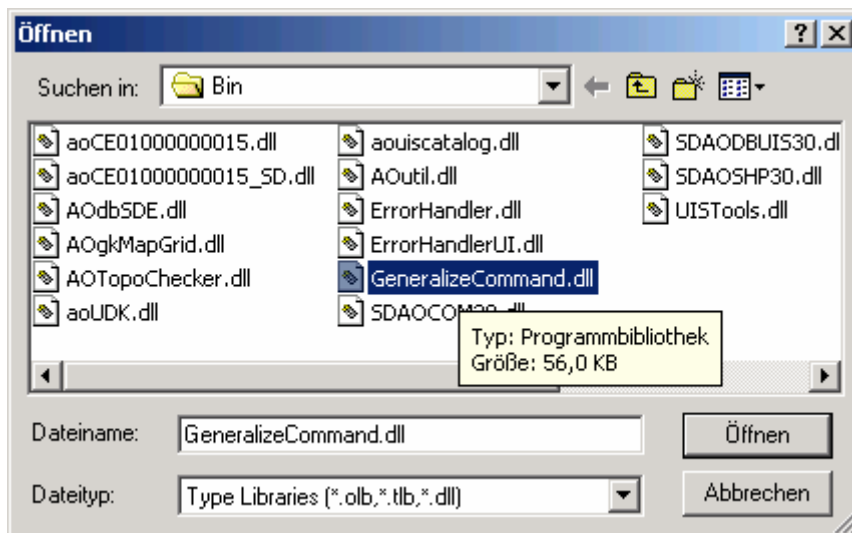


Abb. 45: Direkt öffnen

In dem Programm, in das der Generalisierungsautomat eingebunden werden soll, findet man unter „Tools“ -> „Customize...“ die Möglichkeit den Automaten einzubinden. Man findet ihn unter „GIS_TEAM“. Das Command kann nun in die Menüleiste gezogen werden und von dort aus gestartet werden.

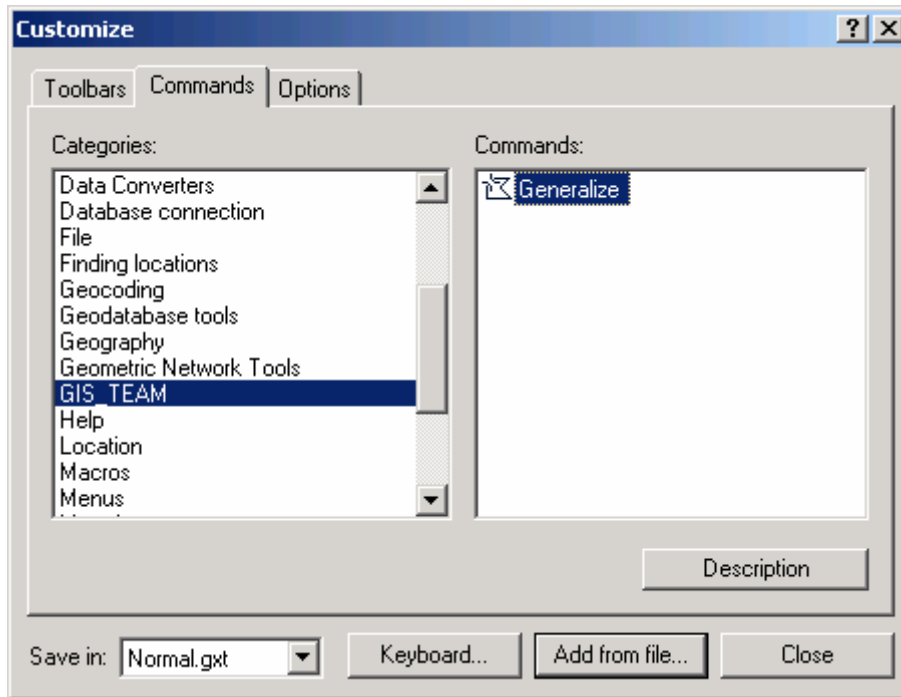


Abb. 46: Customize

6 Die Generalisierung im Test

Die Generalisierung der Daten in Verbindung mit Vektordaten ist unumgänglich, um hohe Qualität zu gewährleisten und damit die Aussagekraft zu erhöhen. Die Karten sollten sich nur auf das Wesentliche beziehen und nicht überladen sein. Dies führt zu einer schnellen Erfassbarkeit des Inhalts, was bei einer Intranet- oder Internetveröffentlichung gegeben sein sollte. Zur Gewährleistung der Datenintegrität zwischen Flächendaten und Geobasisdaten erfolgt eine Aggregation (Zusammenfassung) der Bodendaten und eine maßstabsbedingte Generalisierung. Es gilt Überschneidungen auszuschalten und richtig darzustellen. Je nach Verwendungszweck ist es erforderlich auch stärkere Generalisierungen bereitzustellen. Da beispielsweise die Daten der LSG im Maßstab 1:25.000 vorliegen, müssen diese für die Einbindung in das Intranet mit verschiedenen Maßstäben bis zu 1:3.000.000 generalisiert werden.

Großes Augenmerk fiel auf die Datengröße und die damit verbundene Geschwindigkeiten beim Laden.

Datengrößen:

Toleranz	Generalize	Weed *
Original	8.07 MB	8.07 MB
50	1.44 MB	1.42 MB
100	955 KB	939 KB
200	609 KB	600 KB
300	472 KB	465 KB
500	342 KB	338 KB
1000	219 KB	216 KB

*weed = Toleranz * 1000 (entspricht der Distanz bei generalize)

Tabelle 6: Datengrößen

Da bisher immer nur an Hand der LSGs getestet und gearbeitet wurde, wurden der Generalisierungsautomat auf seine Tauglichkeit und Veträglichkeit mit andern Daten erprobt. Hierfür wurden die Biotop 24a genommen. Die Biotop-Daten bestehen aus 117.523 Polygonen.

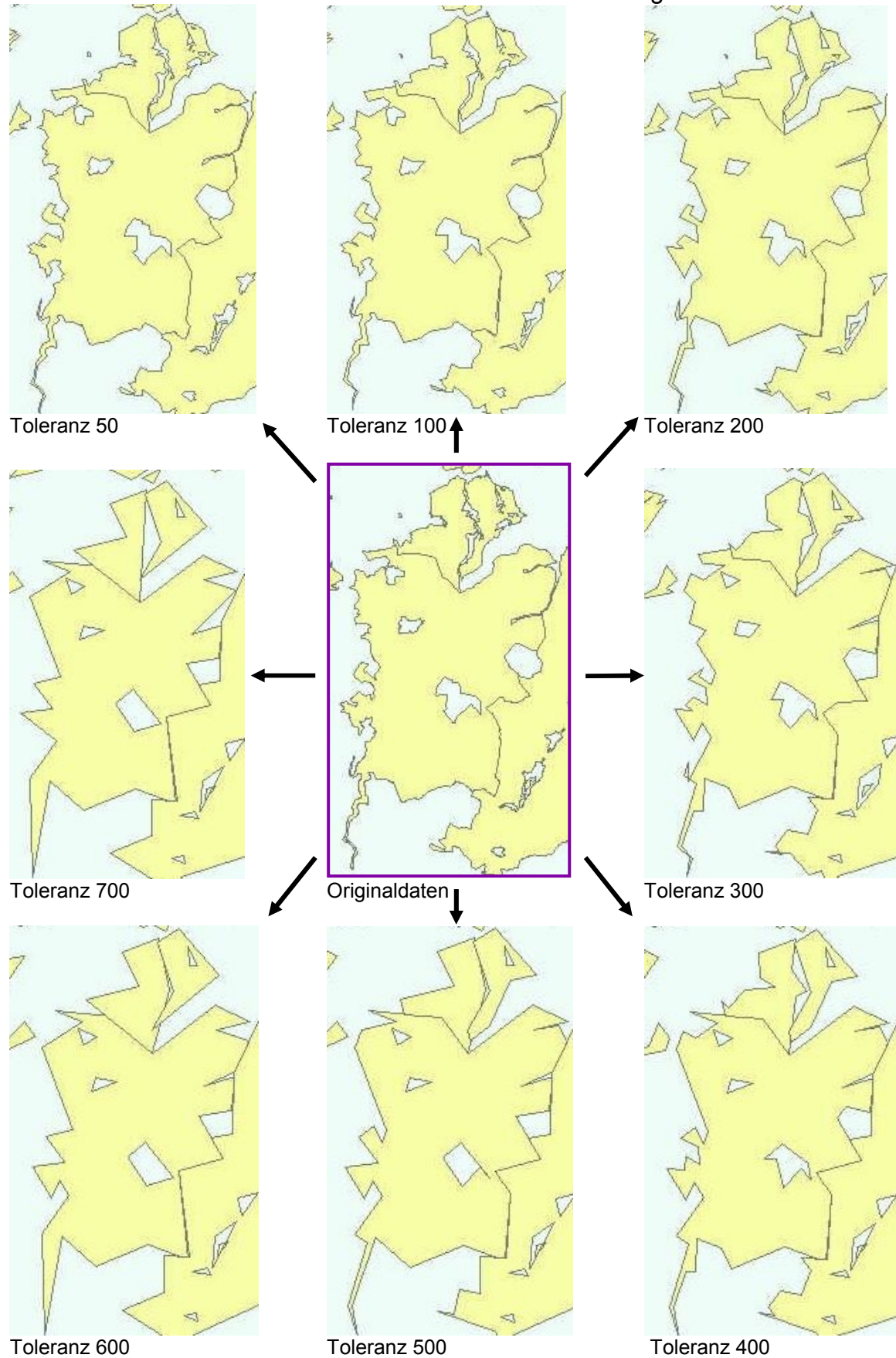
Das Originale-Shapefile hat in ArcView eine Ladezeit von ca. 15 sec. Nach der Bearbeitung mit dem Generalisierungsautomaten, bei der eine Toleranz von 10 und der Generalize-Algorithmus angegeben wurde, hatte sich die Ladezeit auf 5 sec verkürzt!

Des weiteren waren auf Grund der niedrigen Toleranz alle 117.523 Polygone erhalten geblieben und die neue Geometrie unterscheidet sich kaum vom Original. Um Unterschiede feststellen zu können muss in den Maßstab 1:100 gezoomt werden, was der normale Nutzer nicht benötigt.

Das Werkzeug eignet sich also nicht nur zur Generalisierung sondern u.a. auch zur Optimierung der bestehenden Geometrien.

6.1 Generalisierungsbeispiele

Ausschnitte aus den LSG in verschiedenen Generalisierungsstufen:



7 Die Interneteinbindung

Das Web und seine Werkzeuge bieten zahlreiche Vorteile, wie die einfache Benutzung dieser Technik. Zur Dynamik der Seiten trägt ein Internet-Map-Server bei, der die Dokumente immer erst bei ihrer Benötigung anfragt. Dies spart Ladezeiten und hält die Internetseite flexibel. Es werden mit Hilfe des Map-Servers Synthesekarten eingebunden. Ziel der Synthesekarte ist es, schnell eine Übersicht zu geben und gleichzeitig die Möglichkeit zu bieten, nähere Informationen und Zoom-Möglichkeiten zu erhalten. Der Nutzer hat eine einheitliche Sicht auf die unterschiedlichen Datenformate. Um sich diese Daten zu erschließen, muss er weder über die zuvor genannten Systeme verfügen, noch deren Funktionalität beherrschen. Die Daten selbst bleiben beim Anbieter und deren Systemen. Der Nutzer erhält ausschließlich Daten für die Vermittlung der Informationen. Über diese Interaktive Oberfläche hat der Nutzer die Möglichkeit innerhalb der Daten zu zoomen und sich Einzelinformationen zu beschaffen. Es werden die vom Nutzer gewünschten Informationen in einer einheitlichen Form präsentiert. Die Standard-Browser Netscape und Internet Explorer unterstützen die Nutzung auf allen Netzen. Damit entfallen die bisher unüberwindlichen Betriebssystemschranken sowie Unterschiede bei den Graphiksystemen und die kartographischen Informationen lassen sich dadurch nutzen. So können diese Daten als digitale Karten zur Präsentation und Weiterverarbeitung mit Zusatzinformationen bereitgestellt werden.

Der Zugriff zu den Daten ist über den zentralen Geodatenserver im Intranet möglich.

7.1 Zusammenstellen der Daten mit GISterm

GISterm ist ein Geodatenzugriffssystem. Es ermöglicht auf räumlich verteilte Geodaten zuzugreifen und im Intra- oder Internet zur Verfügung zu stellen. Die einzelnen Daten, müssen dafür gesammelt und überarbeitet werden. Um die Karte für das Internet zusammenzustellen, werden unter dem Menüpunkt „Karte“ mehrere Möglichkeiten zur Verfügung gestellt. Gibt es bereits Kartenelemente, die schon als .mml-Karte gespeichert sind, so können diese unter „Kartendefinition laden..“ aufgerufen werden. Sollen bereits bestehende Kartendefinition als Erweiterung zur aktuellen Karte hinzugefügt werden, so ist das unter „Kartendefinition hinzufügen...“ möglich. Sind alle Elemente zusammengestellt und die Eigenschaften bereits bearbeitet, so muss das ganze als .mml-Datei gespeichert werden unter „Kartendefinition speichern...“ So können mit Hilfe von „Kartendefinition laden...“ die als .mml-Dateien abgelegten Einstellungen und Zusammenstellungen wieder aufgerufen werden.

Um die Shapefiles zu laden, die mit dem Generalisierungsautomaten bereits vorbereitet wurden, sind unter der Option „Karten“ -> „Importieren“ -> „Shapefile öffnen...“, die nötigen Schritte geführt.

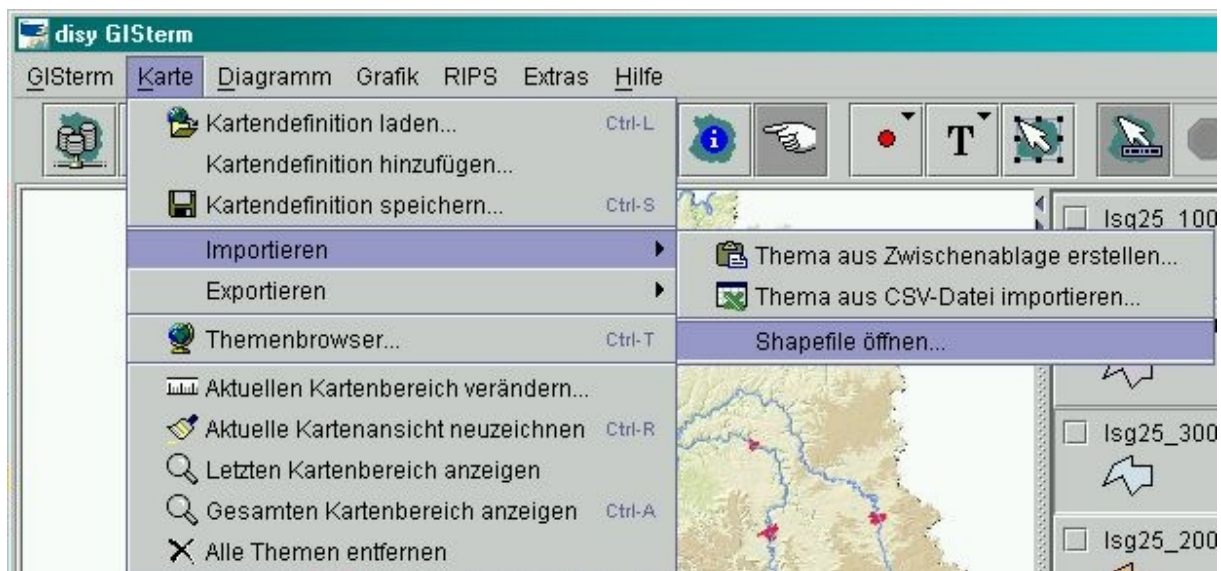


Abb. 56: Shapefiles importieren

Als nächstes öffnet sich die Benutzeroberfläche, die einem Explorer gleich die Daten öffnet. Die geöffneten Daten werden in GISterm an oberster Stelle eingefügt und aktiviert. Jedes geladene Shapefile entspricht einem Layer. Ein Layer ist ähnlich einer Folie und kann individuelle Einstellungen erfahren. Es besteht die Möglichkeit mehrere Layer übereinander zu legen, so z.B. die topographische Karte unter dem sachbezogenen Layer des Shapefiles.

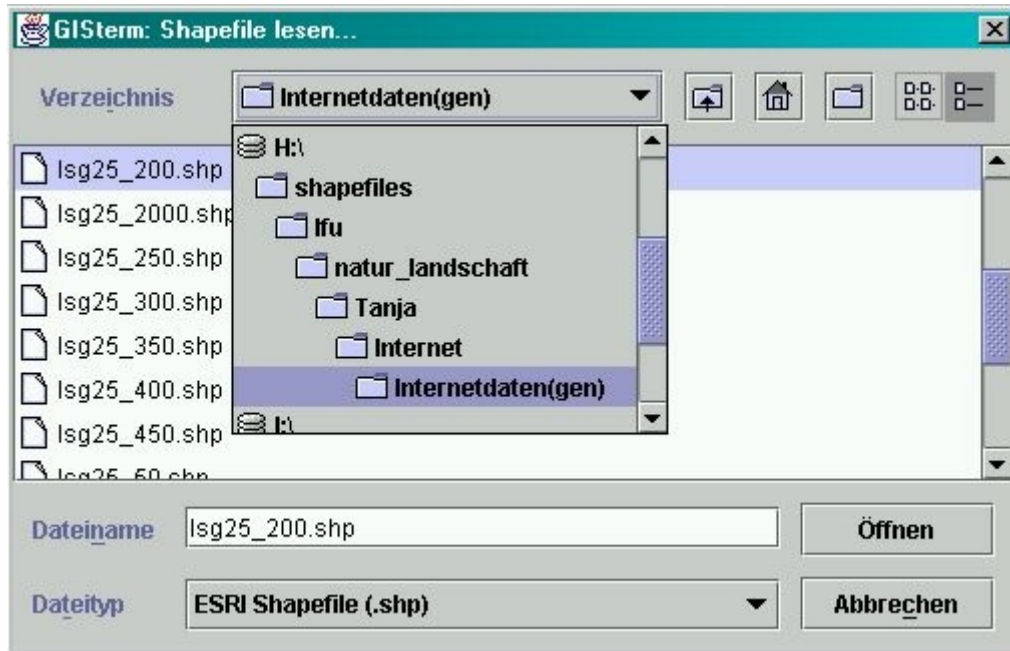


Abb. 57: Shapefile wählen

Um die Daten bearbeiten zu können, müssen diese aktiv sein. Mit der rechten Maustaste geht folgendes Menü auf. Hier können individuell für dieses Shapefile Veränderungen vorgenommen werden, wie z.B. die Position ändern, das Shapefile entfernen, Farbe und Masstäbe verändern usw.



Abb. 58: Eigenschaften der Shapefiles bearbeiten

Ist die Option „Eigenschaften...“ ausgewählt worden, so zeigt sich folgendes Bild.

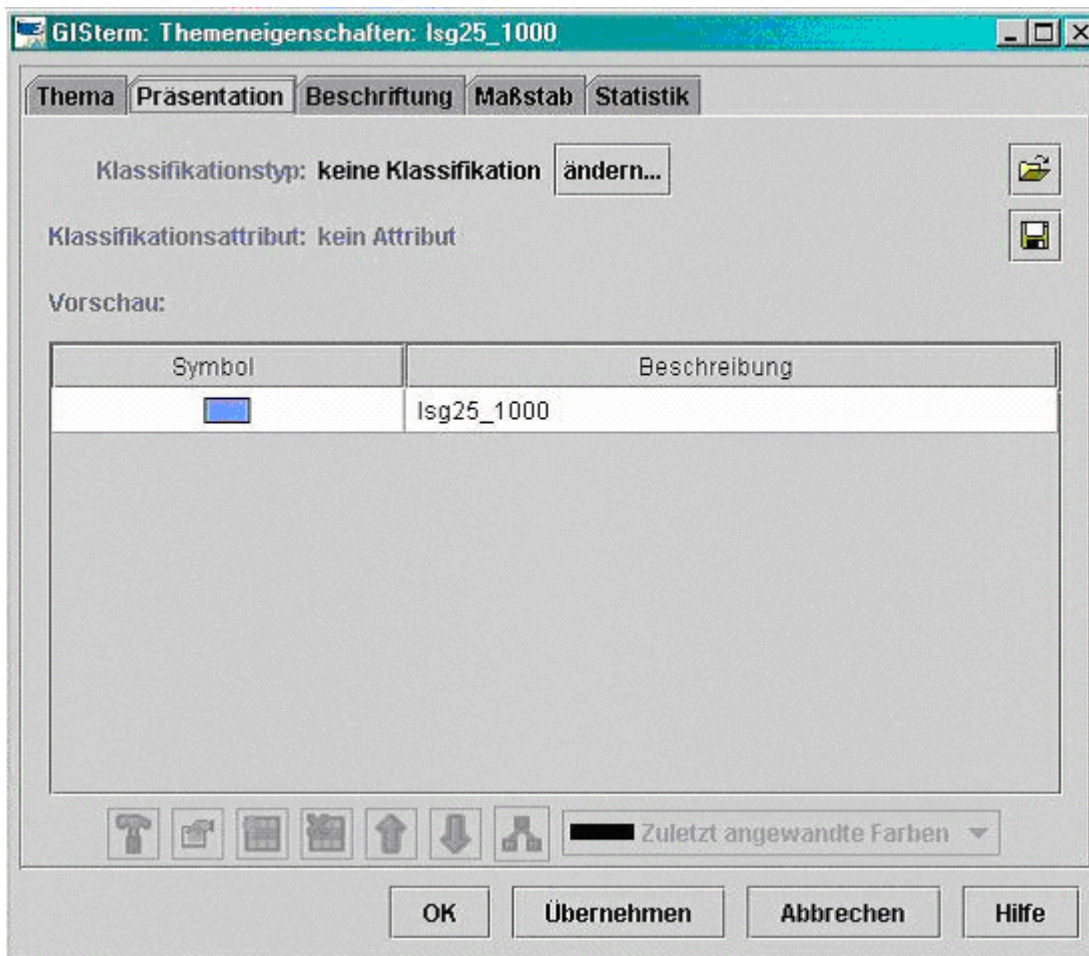


Abb. 59: Eigenschaften Übersicht

Die Farbeinstellungen können mit einem Doppelklick aktiviert werden. Es bestehen die Möglichkeiten den Stil, die Umrandung, die Farbe selbst und die Transparenz zu verändern.

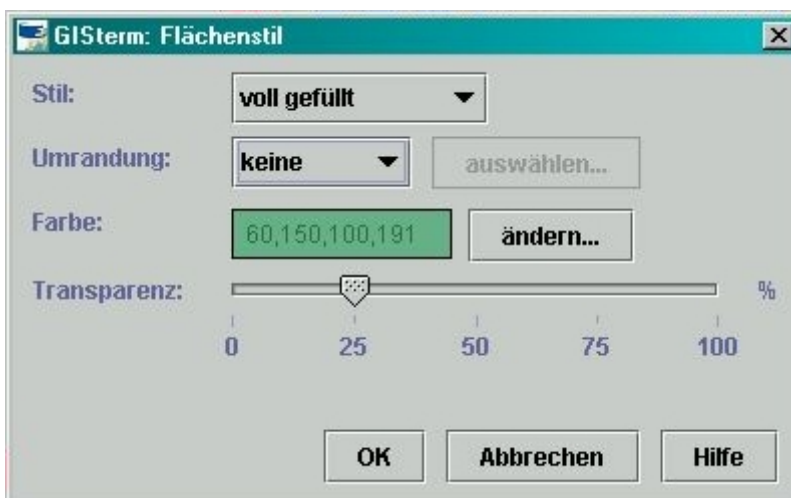


Abb. 60: Farbeinstellungen

Unter „Maßstab“ werden die Gültigkeitsbereiche des Shapefiles festgelegt. Ist der aktuelle Maßstab außerhalb des hier angegebenen Bereichs, so wird das Shapefile ausgeblendet.

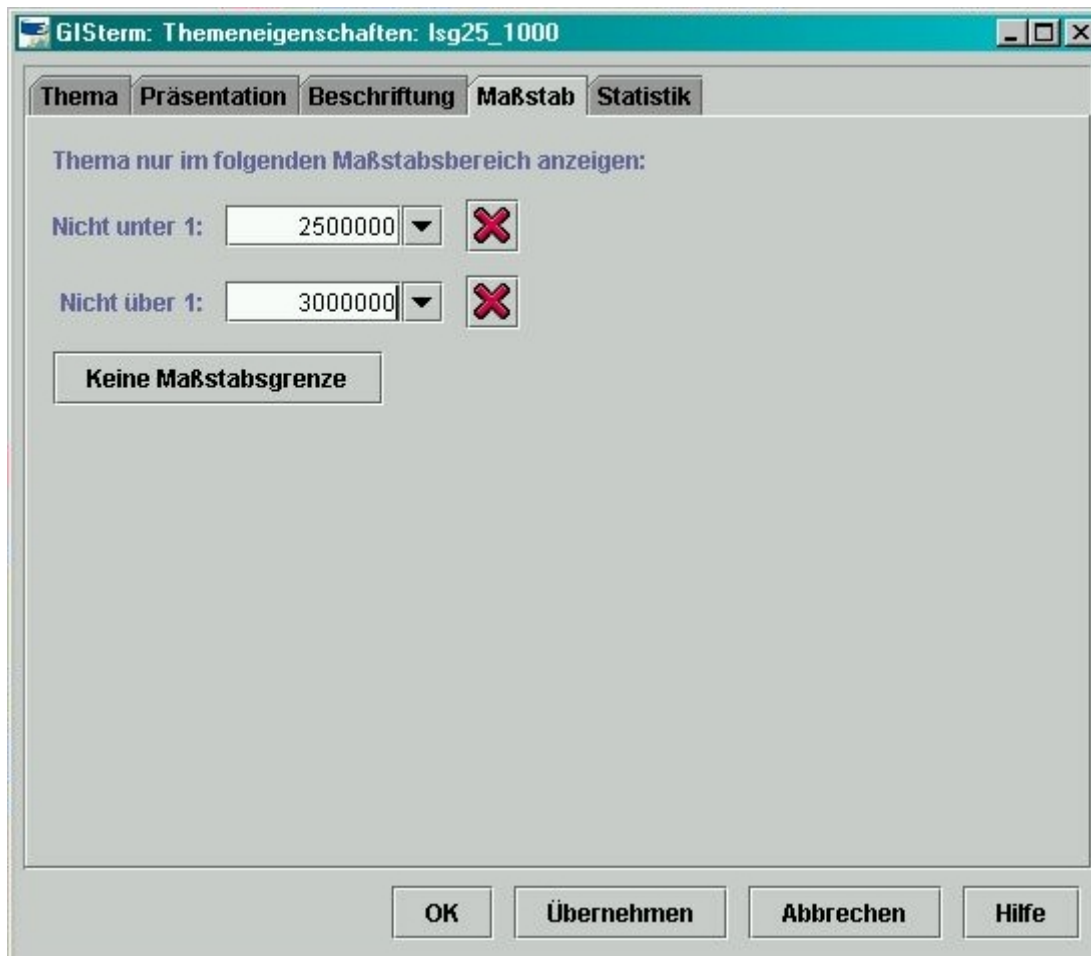


Abb.61: Maßstabseinstellungen

Sind alle Eigenschaften bearbeitet und können in dieser Version in das Internet gestellt werden, muss diese Version abgespeichert werden, wie oben beschrieben unter „Kartendefinition speichern...“. Diese Kartendefinition speichert die Karte einschließlich der Legende in einer mml-Datei und kann somit mit allen Einstellungen im Internet aufgerufen werden. Zuvor sollten allerdings die Einstellungen überprüft werden, damit auch im Internet keine Probleme auftauchen.

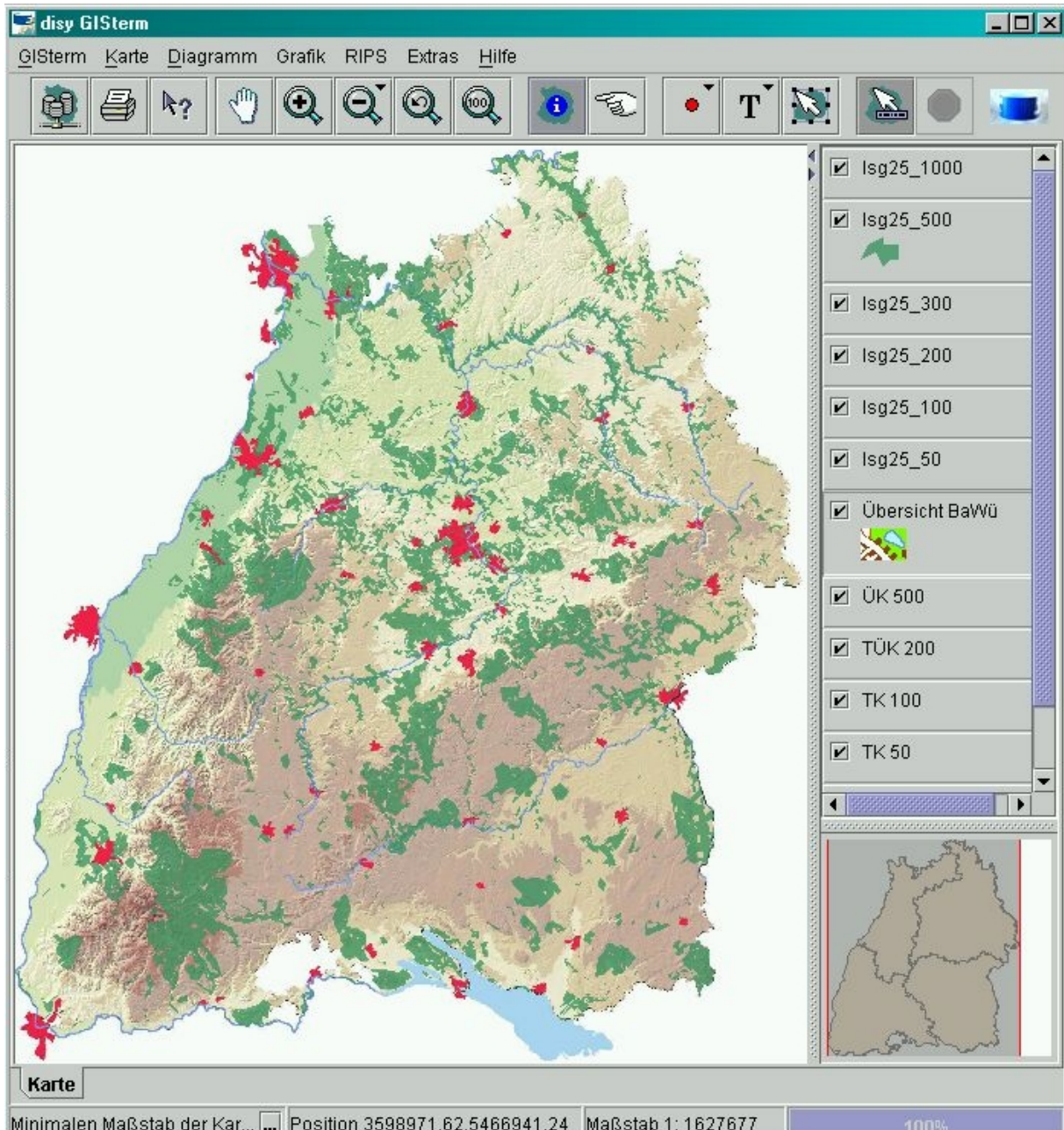


Abb.62: GISterm nach der Einbindung und Bearbeitung aller Tools

7.2 Der Internetaufruf

Die generalisierten LSG-Daten wurden in mml-Dateien namens UdkLsg.mml abgespeichert.

Der Ausschnitt aus der UdkLsg.mml zeigt die Einbindung der lsg25_100.shp. Diese beinhaltet die generalisierten Daten mit dem Toleranzwert 100.

UdkLsg.mml:

[...]

```
<layer enabled="true" editable="false">
  <layerName>lsg25_100</layerName>
  <scale>
    <maxScale>750000.0</maxScale>
    <minScale>250000.0</minScale>
  </scale>
```

] Name des Layers
] Maßstabsangabe min. 25000 bis max. 750000

[...]

```
<URL>[0]lfu/natur_landschaft/Tanja/Internet/Internetdaten/lsg25_100</URL>
  </fileQuery>
  </queryType>
</dataSource>
<legend type="SingleSymbol" attribute="" name="lsg25_100"
description="" classSorting="Unsorted">
  <legendClass name="lsg25_100">
    <color red="60" green="150" blue="100" alpha="191" />
    <symbols>
      <fillSymbol hasOutline="false">
        <simpleFillSymbol style="Solid" />
      </fillSymbol>
      <lineSymbol width="1">
        <simpleLineSymbol style="Solid" />
      </lineSymbol>
      <markerSymbol size="10">
        <simpleMarker style="Circle" />
        <color red="60" green="150" blue="100" alpha="191" />
      </markerSymbol>
```

] Pfad
] Farbfestlegung, so wie weitere Einstellungen

[...]

Alle benötigten Topographischen Karten wurden in der Datei BaWue.mml abgelegt. Auch hier wurden Angaben zu Maßstabsbereichen und individuelle Einstellungen gemacht und gespeichert.

Der Ausschnitt aus der BaWue.mml zeigt die Einbindung der TK50

BaWue.mml:

[...]

```

<layer enabled="true" editable="false">
  <layerName>TK 50</layerName>
  <scale>
    <maxScale>40000.0</maxScale>
    <minScale>20000.0</minScale>
  </scale>
  <dataSource>
    <queryType activeLayer="true">
      <serverQuery>
        <serverConnection>
          <serverName>JDBCRIPSAdapter</serverName>
          <login>GEO_READ</login>
          <password>GEO_READ</password>
        </serverConnection>
        <queryData>
          <geoFeatureType>RIPS@Topographische
Karten@TK50</geoFeatureType>
        </queryData>
      </serverQuery>
    </queryType>
  </dataSource>
</layer>

```

} Name der Layers

} Maßstabsangabe min. 20000 bis max. 40000

Um die Daten im Intranet aufrufen zu können, muss eine .xml-Datei erstellt werden. Diese beinhaltet die Pfade zu den zu ladenden Daten. Für die Veröffentlichung der LSGs müssen die generalisierten LSG-Datei „UdkLsg.mml“ und die topographischen Karten als Grundlage namens „BaWue.mml“ aufgerufen werden.

Ausschnitt aus der .xml-Datei:

[...]

```

<layer select="true">
  <title>Landschaftsschutzgebiete</title>
  <dataSource>bsp_v2:UdkLsg</dataSource>
</layer>
<layer>
  <title>Topographische Karten</title>
  <dataSource>bsp_v2:BaWue</dataSource>
</layer>

```

} Aufruf der LSGs, der UdkLsg.mml

} Aufruf der Topographischen Karten, der BaWue.mml

[...]

Die folgende Abbildung zeigt die Intranetausgabe mit disy. Die Layer können vom Benutzer nach belieben aus- oder zugeschaltet werden. Unter Legende können Informationen über die Karteninhalte einzeln beschrieben eingeholt werden. Die Optionsbuttons über der Karte geben die Möglichkeit innerhalb der Karte zu navigieren. Die Hand symbolisiert „verschieben der Karte“, die Lupen mit + und – ermöglichen das „Zoomen“ und die 3.Lupe das Zurückführen in den Ausgangszusand. Mit dem Informationbutton kann man mit einem Klick an die gewünschte Stelle in der Karte Sachinformationen erhalten. Das Fragezeichen beantwortet technische Fragen bei einem Mausklick.

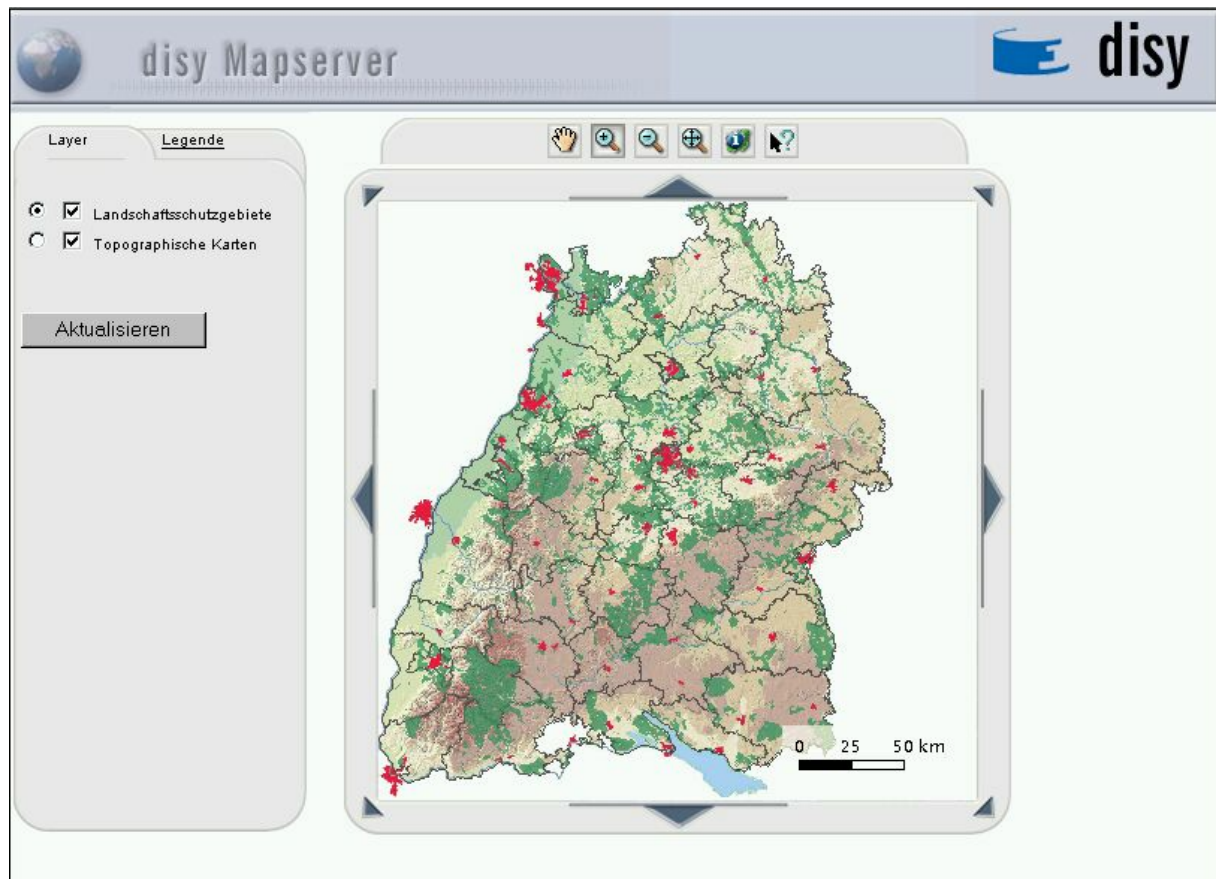


Abb. 63: Intranetdarstellung

8 Zusammenfassung

Die Diplomarbeit beinhaltet mehrere Generalisierungsmethoden. Abschluss ist ein Generalisierungsapparat, der die Daten vor der Abbildung in die gewünschte Form bringt. Die verschiedenen Massstäbe der Abbildungen erfordern verschiedene Inhaltsdichten, so dass eine schnelle Erfassbarkeit der Karteninhalte gegeben ist. Eine Benutzeroberfläche ist geschaffen worden, um dem Nutzer eine möglichst einfache Bedienung der Datenmanipulation ohne Hintergrundwissen zu ermöglichen. Die Präsentation im Intranet besteht aus einer Zusammenfassung der vorher generalisierten Umweltdaten. Hier ist ein weiterer Augenmerk darauf gerichtet worden, welchen Speicherbedarf die Daten benötigen, um unnötige Wartezeiten, während des Ladens zu verringern.

9 Literaturverzeichnis:

Bartelme, Norbert

Geoinformatik: Modelle, Strukturen, Funktionen, 414 S. Berlin, Heidelberg: Springer, 1995.

Braun, Gabriele & Buzin & Wintges (Hrsg.)

GIS und Kartographie im Umweltbereich, 209 S. Heidelberg: Wichmann, 2001.

Hake, Günter & Dietmar Grünreich

Kartographie, 7., völlig neu bearb. u. erw. Aufl., 599 S. Berlin: de Gruyter, 1994.

Herter, Michael & Michael Höck & Michael Jacobi

Avenue TM : Programmierung in ArcView GIS, 414 S. Freising: Eigenverl., 1999.

Höck, Michael & Jochen Manegold

ArcMap Programmierung mit VBA, 758 S. Eigenverlag, 2001

Kemper, Alfons & André Eickler

Datenbanksysteme: eine Einführung, 2., aktualisierte und erw. Aufl., 504 S.

München, Wien: Oldenbourg, 1997.

McCoy, Jill & Kevin Johnston

Using ArcGIS Spatial Analyst, 230 S., ESRI. Eigenverlag, 2001.

Saurer, Helmut & Franz-Josef Behr

Geographische Informationssysteme : eine Einführung ; mit 31 Tabellen , 235 S.

Darmstadt: Wiss. Buchges., 1997.

Tilli, Thomas

Automatisierung mit Fuzzy-Logik: Fuzzy-Hardware und Entwicklungstools im praktischen Einsatz ; mit HD-Diskette, 2., überarb. Aufl., 256 S. München: Franzis, 1993.

Tucker, Corey

Using ArcToolbox, 105 S., ESRI. Eigenverlag, 1999, 2000.

Vienneau, Aleta

Using ArcCatalog, 286 S., ESRI. Eigenverlag, 1999, 2001.

10 Abbildungsverzeichnis

		Seite
1	Rasterformat	10
2	Vektorformat	11
3	ArcGIS	15
4	ESRI	16
5	Value und Count	17
6	GISterm	19
7	MapServer	20
8	Douglas/Peucker-Algorithmus	21
9	Aktivierte Quelldaten	22
10	Menüleiste mit Pop-up „Thema“	22
11	Speichern unter ...	23
12	Einstellungen	23
13	Zu bearbeitendes Feld	23
14	Legende mit neuen Rasterdaten „Bio10“	24
15	Skript „Rasterdaten in Shapes umwandeln“	24
16	Bildschirmübersicht	25
17	Speichern	25
18	Automatisch eingebunden	26
19	Aus Raster wird Shape	26
20	„Verbindung“	27
21	ArcToolbox	28
22	vom Shapefile zur Coverage	29
23	Poly-Coverage zum Grid	29
24	Farbverlauf	30
25	Lösung	31
26	Vorschau einfabig	31
27	Unix-Oberfläche	32
28	Shape in Coverage umwandeln	34
29	Generalisierung	35
30	Clean	35
31	Coverage zu Shapefile	36
32	Werkzeuge verwalten	36
33	„Meine Werkzeuge“ benennen	36
34	Erstellung der Benutzeroberfläche und Toolbox	37
35	Verbindung des Generalize-Buttons mit dem Macro	38
36	Startbutton	39
37	Benutzeroberfläche	40
38	Originaldaten	40
39	Error	41
40	Speichern unter ...	41
41	Prozessorüberwachung	41
42	Abschlussangaben	42
43	Hilfefenster	42
44	Registrierung	43
45	Direkt öffnen	43
46	Customize	44

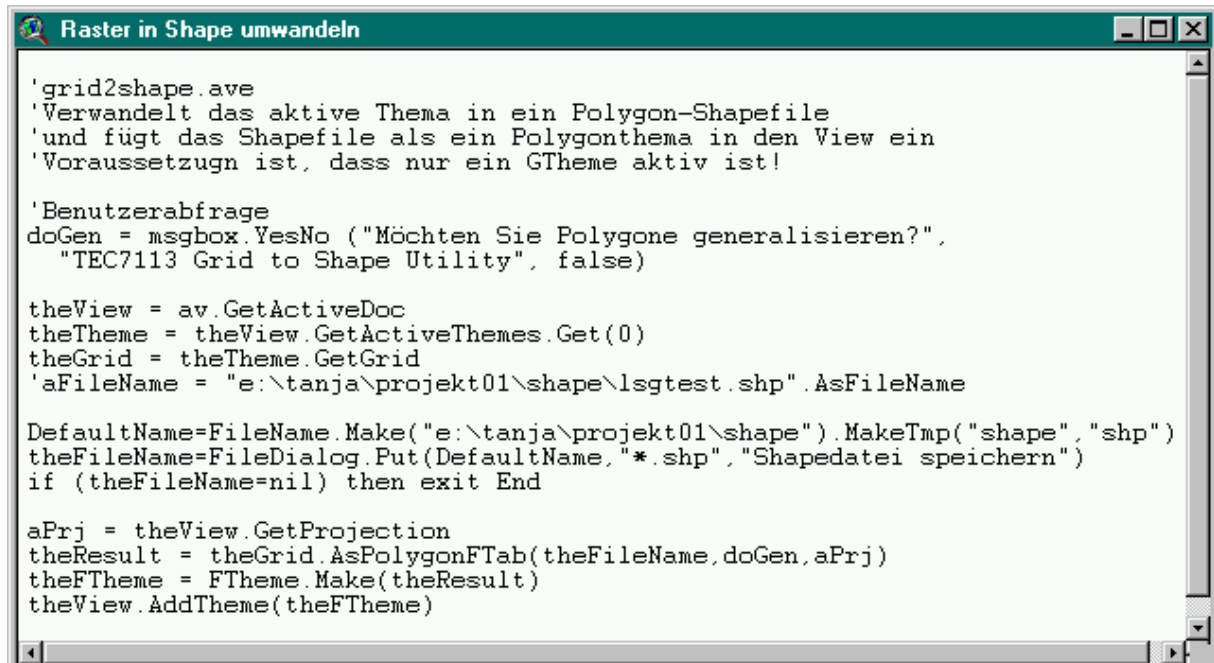
47-55	Generalisierungsbeispiele	46
56	Shapefile importieren	48
57	Shapefile wählen	49
58	Eigenschaften des Shapefiles bearbeiten	49
59	Eigenschaften Übersicht	50
60	Farbeinstellungen	50
61	Maßstabseinstellungen	51
62	GISterm nach der Einbindung und Bearbeitung aller Tools	52
63	Intranetdarstellung	55

11 Tabellenverzeichnis

		Seite
1	mögliche Maßstabsbereiche	8
2	Landschaftsschutzgebiete	12
3	Naturschutzgebiete	13
4	Daten	13
5	Topographische Kartengrundlagen	14
6	Datengröße	45

12 Anhang

Script mit der Sprache Avenue aus ArcView GIS, um Rasterdaten in Shapes umzuwandeln:



```
'grid2shape.ave
'Verwandelt das aktive Thema in ein Polygon-Shapefile
'und fügt das Shapefile als ein Polygonthema in den View ein
'Voraussetzungen ist, dass nur ein GTheme aktiv ist!

'Benutzerabfrage
doGen = msgbox.YesNo ("Möchten Sie Polygone generalisieren?",
    "TEC7113 Grid to Shape Utility", false)

theView = av.GetActiveDoc
theTheme = theView.GetActiveThemes.Get(0)
theGrid = theTheme.GetGrid
'aFileName = "e:\tanja\projekt01\shape\lsgtest.shp".AsFileName

DefaultName=FileName.Make("e:\tanja\projekt01\shape").MakeTmp("shape", "shp")
theFileName=FileDialog.Put(DefaultName, "*.shp", "Shapedatei speichern")
if (theFileName=nil) then exit End

aPrj = theView.GetProjection
theResult = theGrid.AsPolygonFTab(theFileName,doGen,aPrj)
theFTheme = FTheme.Make(theResult)
theView.AddTheme(theFTheme)
```

Befehlsfolge in ArcGIS um die Biotope zu generalisieren:

```
Arc: shapearc bio24a bioorg bio
Arc: list bioorg.patbio
Arc: build bioorg poly
Arc: clean bioorg bioorg
Arc: generalize bioorg bio1000v 1000 bendsimplify
Arc: build bio1000v poly
Arc: clean bio1000v bio1000v
Arc: arcshape bio1000v regions.bio bio1000v
```

Der DOUGLAS/PEUCKER-Algorithmus, zur Generalisierung von Shapefiles,
Beschreibung siehe Text:

```
theView = av.GetActiveDoc  
thePrj = theView.GetProjection
```

```
if (thePrj.IsNull) then  
  hasPrj = false  
else  
  hasPrj = true  
  thePrj = theView.GetProjection  
end
```

```
'Get active themes
```

```
'-----
```

```
theThemes = theView.GetActiveThemes
```

```
if (theThemes.Count < 1) then
```

```
  MsgBox.Error("Please make a polyline or polygon theme active", "**** Error ****") exit  
end
```

```
' Process each active theme
```

```
'-----
```

```
for each t in theThemes
```

```
' Check that the theme is of the correct type
```

```
'-----
```

```
if (t.Is( FTHEME).Not) then
```

```
  MsgBox.Error (t.GetName++"is not a polyline or polygon theme", "**** Error ****")  
  continue  
end
```

```
theClassName = t.GetFTab.GetShapeClass.GetClassName
```

```
if ((theClassName = "Polygon").Not and (theClassName = "Polyline").Not) then
```

```
  MsgBox.Error (t.GetName ++"is not a polyline or polygon theme", "**** Error ****")  
  continue  
end
```

```
' Get a tolerance in (distance units) to be used (must be > 0)
```

```
'-----
```

```
theTol = 0
```

```
theMUnits = theView.GetDisplay.GetUnits
```

```
theDUnits = theView.GetDisplay.GetDistanceUnits
```

```
theUStr = Units.GetFullUnitString(theDUnits).LCase
```

(zu Douglas/Peucker-Algorithmus)

```
while ((theTol <= 0) or (theTol.AsString.IsNumber).Not)
theTol = MsgBox.Input("Positive tolerance value (in "+theUStr+ ") for
theme"++t.GetName,"Tolerance",theTol.AsString)

if (theTol = NIL) then
    exit
end

if (theTol.IsNumber) then
    theTol = theTol.AsNumber
end
end

theTol = Units.Convert(theTol,theDUnits,theMUnits)

' Get the name for a new theme, from View.Export script
' -----
def = av.GetProject.MakeFileName("_"+t.GetName, "shp")
def = FileDialog.Put(def, "*.shp", "New file for simplified "+ t.getName)

if (def = NIL) then
    return nil
end

' Create a new shapefile theme, from View.Export script
' -----
tbl = t.GetFtab
shpfld = (tbl.FindField("Shape"))

if (shpfld.IsVisible.Not) then
    shpfld.SetVisible(shpfld.IsVisible.Not)
    WasNotVisible = TRUE
    else
        WasNotVisible =FALSE
    end
end

' This will only export the selected shapes, if any have been selected
' -----

av.ShowMsg("Creating new shape file from"++ t.GetName)
anFTab = tbl.Export(def, Shape, tbl.GetSelection.Count > 0) anFTab.SetEditable(true)

if (WasNotVisible) then
    shpfld.SetVisible(FALSE)
end

' Find the shape field
theSField = anFTab.FindField("Shape")
```

(zu Douglas/Peucker-Algorithmus)

```
' Set up status bar
av.ShowMsg("Generalizing new shapes from theme"++ t.GetName)
theSize = anFTab.GetNumRecords
```

```
' Initialize counters for reporting
count = 0
oldCount = 0
newCount = 0
```

```
' Process each shape
'-----
```

```
for each r in anFTab
```

```
av.SetStatus(100*count/theSize)
```

```
' Retrieve shape
'-----
```

```
if (theClassName = "Polygon") then
  theOShape = Polygon.MakeNull
else
  theOShape = Polyline.MakeNull
end
```

```
anFTab.QueryShape(r,theView.GetProjection,theOShape)
```

```
' Make a list for collecting the new set of point lists
theNShape = List.Make
```

```
' Process each list in the ListofListofPoints
'-----
```

```
for each p in theOShape.AsList
```

```
' Make a list for collecting the vertices to keep
theNList = List.Make
oldCount = oldCount + p.Count
```

```
' Set up stack for holding a set of lists of form {index,point}
theStack = Stack.Make
```

(zu Douglas/Peucker-Algorithmus)

' Add first point to the list being assembled, and make it the anchor
theAnchor = p.Get(0)

if (hasPrj) then
 theNList.Add(theAnchor.ReturnUnProjected(thePrj))
else
 theNList.Add(theAnchor)
end

aIndex = 0

' Add last point to stack

'-----
fIndex = p.Count - 1
theStack.Push({fIndex,p.Get(fIndex)})

' Process the points with the Douglas - Peucker approach

'-----

while (theStack.IsEmpty.Not)

' Process from Anchor (beginning) to Floater (top of stack)
fIndex = theStack.Top.Get(0)

' Create graphic shape for comparing distances
theVect = Line.Make(theAnchor,theStack.Top.Get(1))

' Initialize values for comparison
theMax = theTol
mxIndex = 0

' Find distance

'-----

for each i in (aIndex + 1) .. (fIndex - 1)

' process middle points
dist = theVect.Distance(p.Get(i))

' Check against previous maximum
if (dist >= theMax) then
 ' this point is out of the tolerance
 theMax = dist
 mxIndex = i

end

end

(zu Douglas/Peucker-Algorithmus)

' If a vertex is found outside the current tolerance corridor,
' push it onto the stack

if (mxIndex > 0) then

 'a new floater has been found

 theStack.Push({mxIndex,p.Get(mxIndex)})

else

 ' add floater to the list for the new shape and move forward

 if (hasPrj) then

 theNList.Add(theStack.Top.Get(1).ReturnUnProjected(thePrj))

 else

 theNList.Add(theStack.Top.Get(1))

 end

 ' Make the floater the new anchor

 theAnchor = theStack.Pop.Get(1)

 aIndex = fIndex

end

end

' Check for collapsed polygons

'-----

if ((theNList.Count < 4) and (theClassName = "Polygon")) then

 ' Revert to original polygon

 '-----

 if (hasPrj) then

 theNList = List.Make

 for each pt in p

 theNList.Add(pt.ReturnUnProjected(thePrj))

 end

 else

 theNList = p

 end

end

(zu Douglas/Peucker-Algorithmus)

```
' Finish up individual lists
'-----
newCount = newCount + theNList.Count
theNShape.Add(theNList)
end

' Finish up the shape
'-----
if (theClassName = "Polygon") then
  theNShape = Polygon.Make(theNShape)
else
  theNShape = Polyline.Make(theNShape)
end

anFTab.SetValue(theSField, r, theNShape)
count = count + 1
end

' Clear status message
av.ClearMsg av.ClearStatus

' Report on counts of old and new theme starting enhancement... :-)
'-----
oln = 0
for each i in tbl
  oln = oln + (tbl.returnValue (shpfld, i).returnlength)
end

nln = 0
for each i in anFTab
  nln = nln + (anFTab.returnValue(theSField, i).returnlength)
end

' finished "enhanced" - feature... ;-]
'-----
MsgBox.Info("The old theme with"++ oldCount.AsString ++
"vertices ("+oln.asstring+" Display-Units) has been generalized to"++
newCount.AsString++"vertices ("+nln.asstring+" Display-Units).", "Results")

' Create a theme and add it to the View
'-----
anFTab.SetEditable(false)
fthm = FTheme.Make(anFTab)
theView.AddTheme(fthm)
end

' Bring the View to the front
theView.GetWin.Activate
```

VB (VisualBasic) Marco zum Generalisierungsautomaten:

//->frmGeneralize:

Option Explicit

Public Enum genMethod

 genGeneralize = 0

 genWeed = 1

End Enum

Private m_pGxDialog As IGxDialog

Private m_pGxObjectFilter As IGxObjectFilter

Public Sub Clean(ByVal lDistance As Long, ByVal useGenMethod As genMethod)

 Dim pFeature As IFeature

 Dim pInFeatureClass As IFeatureClass

 Dim pFeatureCursor As IFeatureCursor

 Dim pGeometry As IGeometry

 Dim pInsertFeatureBuffer As IFeatureBuffer

 Dim pInsertFeatureCursor As IFeatureCursor

 Dim pOutFeatureClass As IFeatureClass

 Dim pProgressDlgFact As IProgressDialogFactory

 Dim pProgressDialog As IProgressDialog2

 Dim pStepProgressor As IStepProgressor

 Dim pTopoOperator As ITopologicalOperator2

 Dim pTrackCancel As ITrackCancel

 Dim bContinue As Boolean

 Dim lFeatureCount As Long

 Dim lTotalFeatureCount As Long

 Dim lEmptyFeatureCount As Long

 Dim sFinalMessage As String

 On Error GoTo ErrorHandler

 Set pInFeatureClass = GetShapefile

 If pInFeatureClass Is Nothing Then

 MsgBox "Error selecting Shapefile. Exiting."

 Exit Sub

 End If

 'Exit if featureclass has no shapes

 lTotalFeatureCount = pInFeatureClass.FeatureCount(Nothing)

 If lTotalFeatureCount = 0 Then

 MsgBox "No features found in shapefile. Exiting"

 Exit Sub

 End If

(Zu VB Macro)

```
'Create a new Shapefile
Set pOutFeatureClass = CreateNewShapefile(pInFeatureClass)
If pOutFeatureClass Is Nothing Then
    MsgBox "Error creating new Shapefile, check folder permissions."
    Exit Sub
End If
```

```
'Show a progress dialog while we cycle through the features
Set pTrackCancel = New CancelTracker
Set pProgressDlgFact = New ProgressDialogFactory
Set pProgressDialog = pProgressDlgFact.Create(pTrackCancel, 0)
pProgressDialog.CancelEnabled = True
pProgressDialog.Title = "Exporting and Cleaning Shapefile"
pProgressDialog.Animation = esriProgressGlobe
bContinue = True
```

```
'Set the properties of the Step Progressor
Set pStepProgressor = pProgressDialog
pStepProgressor.MinRange = 0
pStepProgressor.MaxRange = ITotalFeatureCount
pStepProgressor.StepValue = 1
```

```
'Create an insert cursor
Set pInsertFeatureCursor = pOutFeatureClass.Insert(True)
Set pInsertFeatureBuffer = pOutFeatureClass.CreateFeatureBuffer
```

```
'Loop through all features in the feature class, correcting each one,
'and write it out to the new shapefile
Set pFeatureCursor = pInFeatureClass.Search(Nothing, False)
Set pFeature = pFeatureCursor.NextFeature
Do While Not pFeature Is Nothing
    'Update progress dialog
    IFeatureCount = IFeatureCount + 1
    pStepProgressor.Message = IFeatureCount & " of " & ITotalFeatureCount & " Features
processed"
    'Stop processing features if 'Cancel' button is selected
    bContinue = pTrackCancel.Continue
    pStepProgressor.Step
    If Not bContinue Then Exit Do
```

```
'If the feature has an invalid shape, create a new empty one
If pFeature.Shape Is Nothing Then
    Set pFeature.Shape = CreateNewGeometry(pOutFeatureClass)
End If
```

(Zu VB Macro)

```
'Simplify each feature and insert into new feature class
Set pTopoOperator = pFeature.Shape
pTopoOperator.IsKnownSimple = False
pTopoOperator.Simplify
```

```
Dim aPolygon As IPolygon
Dim aGeometry As IGeometry
Set aGeometry = pFeature.Shape
Set aPolygon = aGeometry
```

```
Select Case useGenMethod
    Case genMethod.genGeneralize
        aPolygon.Generalize lDistance
    Case genMethod.genWeed
        aPolygon.Weed lDistance
End Select
```

```
InsertFeature pInsertFeatureCursor, pInsertFeatureBuffer, pFeature, pTopoOperator
'Count number of empty features
Set pGeometry = pTopoOperator
If pGeometry.IsEmpty Then
    lEmptyFeatureCount = lEmptyFeatureCount + 1
End If
'Retrieve next feature
Set pFeature = pFeatureCursor.NextFeature
Loop
```

```
pProgressDialog.HideDialog
```

```
Set pInsertFeatureBuffer = Nothing
Set pInsertFeatureCursor = Nothing
```

```
'Recreate indexes on new Shapefile
CreateIndexes pInFeatureClass, pOutFeatureClass
```

```
'Create summary report message
If bContinue Then
    sFinalMessage = "Operation completed successfully." & vbCrLf & vbCrLf
Else
    sFinalMessage = "Job cancelled." & vbCrLf & vbCrLf
End If
sFinalMessage = sFinalMessage & lFeatureCount & " Features processed." & vbCrLf
If Not lEmptyFeatureCount = 0 Then
    sFinalMessage = sFinalMessage & vbCrLf & lEmptyFeatureCount & " Features were found
to have no shape."
End If
```

(Zu VB Macro)

```
MsgBox sFinalMessage
```

```
Set m_pGxObjectFilter = Nothing
```

```
Set m_pGxDialog = Nothing
```

```
Exit Sub 'Exit to avoid error handler
```

```
ErrorHandler:
```

```
MsgBox "An unexpected error occurred." & vbCrLf & vbCrLf & _  
IFeatureCount & " Features processed." & vbCrLf
```

```
End Sub
```

```
Private Function GetShapefile() As IFeatureClass
```

```
Dim pEnumGxObject As IEnumGxObject
```

```
Dim pFeatureClass As IFeatureClass
```

```
Dim pGxDataset As IGxDataset
```

```
On Error GoTo ErrorHandler
```

```
'Have the user select a shapefile
```

```
Set m_pGxDialog = New GxDialog
```

```
Set m_pGxObjectFilter = New GxFilterShapefiles
```

```
Set m_pGxDialog.ObjectFilter = m_pGxObjectFilter
```

```
m_pGxDialog.Title = "Select a Shapefile to Clean:"
```

```
If m_pGxDialog.DoModalOpen(0, pEnumGxObject) Then
```

```
    pEnumGxObject.Reset
```

```
    Set pGxDataset = pEnumGxObject.Next
```

```
    Set pFeatureClass = pGxDataset.Dataset
```

```
End If
```

```
Set GetShapefile = pFeatureClass
```

```
Exit Function
```

```
ErrorHandler:
```

```
Set GetShapefile = Nothing
```

```
End Function
```

```
Private Function CreateNewShapefile(pInFeatureClass As IFeatureClass) As IFeatureClass
```

```
Dim pClone As IClone
```

```
Dim pFeatureWorkspace As IFeatureWorkspace
```

```
Dim pFields As IFields
```

```
Dim pGxFile As IGxFile
```

```
Dim pNewFeatureClass As IFeatureClass
```

```
Dim pWorkspaceFactory As IWorkspaceFactory
```

```
On Error GoTo ErrorHandler
```

(Zu VB Macro)

```
m_pGxDialog.Title = "Enter New Output Shapefile:"
If m_pGxDialog.DoModalSave(0) Then
    Set pGxFile = m_pGxDialog.FinalLocation
Else
    Set CreateNewShapefile = Nothing
    Exit Function
End If

Set pWorkspaceFactory = New ShapefileWorkspaceFactory
Set pFeatureWorkspace = pWorkspaceFactory.OpenFromFile(pGxFile.Path, 0)
Set pClone = pInFeatureClass.Fields
Set pFields = pClone.Clone
Set pNewFeatureClass = pFeatureWorkspace.CreateFeatureClass(m_pGxDialog.Name,
pFields, Nothing, Nothing, esriFTSimple, pInFeatureClass.ShapeFieldName, "")
Set CreateNewShapefile = pNewFeatureClass

Exit Function

ErrorHandler:
Set CreateNewShapefile = Nothing

End Function

Private Sub InsertFeature(pInsertFeatureCursor As IFeatureCursor, pInsertFeatureBuffer As
IFeatureBuffer, pOrigFeature As IFeature, pGeometry As IGeometry)
    Dim pFields As IFields
    Dim pField As IField
    Dim pPoint As IPoint
    Dim FieldCount As Integer

    'Copy the attributes of the orig feature the new feature
    Set pFields = pOrigFeature.Fields
    For FieldCount = 0 To pFields.FieldCount - 1 'skip OID and geometry
        Set pField = pFields.Field(FieldCount)
        If Not pField.Type = esriFieldTypeGeometry And Not pField.Type = esriFieldTypeOID _
            And pField.Editable Then
            pInsertFeatureBuffer.Value(FieldCount) = pOrigFeature.Value(FieldCount)
        End If
    Next FieldCount

    Set pInsertFeatureBuffer.Shape = pGeometry
    pInsertFeatureCursor.InsertFeature pInsertFeatureBuffer

End Sub
```

(Zu VB Macro)

```
Private Sub CreateIndexes(pInFeatureClass As IFeatureClass, pOutFeatureClass As
IFeatureClass)
    Dim pClone As IClone
    Dim pOutIndexes As IIndexes
    Dim pIndex As IIndex
    Dim pNewIndex As IIndex
    Dim iIndexCount As Integer
    Dim pFields As IFields

    Set pClone = pInFeatureClass.Indexes
    Set pOutIndexes = pClone.Clone

    For iIndexCount = 0 To pOutIndexes.IndexCount - 1
        Set pNewIndex = pOutIndexes.Index(iIndexCount)
        Set pFields = pNewIndex.Fields
        pOutFeatureClass.AddIndex pNewIndex
    Next iIndexCount

End Sub

Private Function CreateNewGeometry(pFeatureClass As IFeatureClass) As IGeometry
    Select Case pFeatureClass.ShapeType
        Case esriGeometryPoint
            Set CreateNewGeometry = New Point
        Case esriGeometryMultipoint
            Set CreateNewGeometry = New Multipoint
        Case esriGeometryPolyline
            Set CreateNewGeometry = New Polyline
        Case esriGeometryPolygon
            Set CreateNewGeometry = New Polygon
    End Select
End Function

Private Sub cmdClose_Click()
    Unload frmGeneralize
End Sub
```


(Zu VB Macro)

```
Private Sub cmdGeneralize_Click()  
    Dim lDistance As Long  
    Dim aGenMethod As genMethod  
  
    If IsNumeric(txtTolerance.Text) Then  
        lDistance = CLng(txtTolerance.Text)  
  
        If optGen.Value Then  
            aGenMethod = genGeneralize  
        Else  
            aGenMethod = genWeed  
        End If  
  
        Clean lDistance, aGenMethod  
    Else  
        MsgBox "Bitte geben Sie eine gültige Distanz an!", vbInformation, "Achtung"  
    End If  
  
End Sub  
  
Private Sub optGen_Click()  
    optWeed.Value = Not optGen.Value  
  
End Sub  
  
Private Sub optWeed_Click()  
    optGen.Value = Not optWeed.Value  
End Sub
```

Intranetaufruf Quellcode: mapClientConfigUdkLsg.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE disyMapServer PUBLIC "-//disy Information
Systems//MapMarkupLanguage//EN"
'file:/usr/local/opt/jakarta/webapps/GIStermMapServer/config/MapClient.dtd'
>
<mapClient>
  <map hasLayerManager="true">
    <wmtVersion>1.0</wmtVersion>
    <mapServletHttp>

<url>http://itz3as.lfu.bwl.de/GIStermMapServer/mapserver/MapServlet</url>
    </mapServletHttp>
    <mapSize width="400" height="400" />
    <mapFormat>GIF</mapFormat>
  </map>
  <legend>
    <legendBgColor>0xECEBE9</legendBgColor>
    <legendTemplate>

<url>file:///usr/local/opt/jakarta/webapps/GIStermMapServer/client/rvnsw/Le
gendBase.jsp</url>
    </legendTemplate>
    <legendResult>

<url>file:///usr/local/opt/jakarta/webapps/GIStermMapServer/client/rvnsw/Le
gende.jsp</url>
    </legendResult>
  </legend>
  <infoJspHttp>
    <!--
<url>itz3as.lfu.bwl.de/GIStermMapServer/client/jsp/MetaData.jsp</url> -->
  </infoJspHttp>
  <infoFormat>XML</infoFormat>
  <externalServletHttp>
    <url />
  </externalServletHttp>
  <layerManager version="1.0">
    <configLocation>
      <url
dirPrefix="007">/usr/local/opt/jakarta/webapps/GIStermMapServer/testkarten<
/urL>
    </configLocation>
    <boundingBox>
      <upperLeftX>3387950.0</upperLeftX>
      <upperLeftY>5518620.0</upperLeftY>
      <lowerRightX>3616648.0</lowerRightX>
      <lowerRightY>5260260.0</lowerRightY>
    </boundingBox>
    <layer select="true">
      <title>Landschaftsschutzgebiete</title>
      <dataSource>bsp_v2:UdkLsg</dataSource>
    </layer>
    <layer>
      <title>Topographische Karten</title>
      <dataSource>bsp_v2:BaWue</dataSource>
    </layer>
  </layerManager>
</mapClient>
```

Intranetzusammenstellung der topographischen Karten: BaWue.mml

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE map PUBLIC "-//disy Information Systems//MapMarkupLanguage//EN"
'file:/C:/UIS/Dienste/GISterm/etc/MML.dtd'>

<map>
  <version>1.0</version>
  <mapModel>
    <boundingBox>
      <upperLeftX>3387950.0</upperLeftX>
      <upperLeftY>5518620.0</upperLeftY>
      <lowerRightX>3736941.0</lowerRightX>
      <lowerRightY>5260260.0</lowerRightY>
    </boundingBox>
    <maxAreaOfInterest>
      <upperLeftX>3388300.0</upperLeftX>
      <upperLeftY>5517400.0</upperLeftY>
      <lowerRightX>3611900.2</lowerRightX>
      <lowerRightY>5258200.0</lowerRightY>
    </maxAreaOfInterest>
    <layer enabled="true" editable="false">
      <layerName>Übersicht BaWü</layerName>
      <scale>
        <maxScale>3000000.0</maxScale>
        <minScale>800000.0</minScale>
      </scale>
      <dataSource>
        <queryType activeLayer="false">
          <fileQuery>
            <fileType>rasterFileQuery</fileType>
            <URL>[1]Uebersicht_BaWue.jpg</URL>
          </fileQuery>
        </queryType>
      </dataSource>
    </layer>
    <layer enabled="true" editable="false">
      <layerName>Kreise</layerName>
      <scale>
        <maxScale>3000000.0</maxScale>
        <minScale>800000.0</minScale>
      </scale>
      <label layerIsLabeled="false" labelIsScaled="true">
        <labelAttr>LANGNAME</labelAttr>
        <labelThreshold>6.0</labelThreshold>
        <labelAnchor>center</labelAnchor>
        <labelRefScale>1200000.0</labelRefScale>
        <fontSize>10</fontSize>
        <fontName>sansserif</fontName>
        <fontStyle>0</fontStyle>
        <tooltipAttr>LANGNAME</tooltipAttr>
      </label>
    </layer>
  </mapModel>
</map>
```

(Zu BaWue.mml)

```
</label>
<dataSource>
  <queryType activeLayer="false">
    <fileQuery>
      <fileType>shapeFileQuery</fileType>
      <URL>[0]/lfu/admin/krei1000</URL>
    </fileQuery>
  </queryType>
</dataSource>
<legend type="SingleSymbol" attribute="" name="Kreise" description=""
classSorting="Unsorted">
  <legendClass name="Kreise">
    <color red="150" green="245" blue="90" alpha="0" />
    <symbols>
      <fillSymbol hasOutline="true">
        <simpleFillSymbol style="Solid" />
        <color red="150" green="245" blue="90" alpha="0" />
        <outline>
          <color red="85" green="85" blue="85" alpha="255" />
          <lineSymbol width="1">
            <simpleLineSymbol style="Solid" />
          </lineSymbol>
        </outline>
      </fillSymbol>
      <lineSymbol width="1">
        <simpleLineSymbol style="Solid" />
      </lineSymbol>
      <markerSymbol size="10">
        <simpleMarker style="Circle" />
        <color red="150" green="245" blue="90" alpha="0" />
      </markerSymbol>
    </symbols>
  </legendClass>
<undefClass name="unclassified">
  <color red="128" green="0" blue="128" alpha="255" />
  <symbols>
    <fillSymbol hasOutline="true">
      <simpleFillSymbol style="Solid" />
      <color red="150" green="245" blue="90" alpha="0" />
      <outline>
        <color red="0" green="0" blue="0" alpha="255" />
        <lineSymbol width="1">
          <simpleLineSymbol style="Solid" />
        </lineSymbol>
      </outline>
    </fillSymbol>
  </symbols>
</undefClass>
</legend>
```

(Zu BaWue.mml)

```
</layer>
<layer enabled="true" editable="false">
  <layerName>ÜK1000</layerName>
  <scale>
    <maxScale>800000.0</maxScale>
    <minScale>400000.0</minScale>
  </scale>
  <dataSource>
    <queryType activeLayer="true">
      <serverQuery>
        <serverConnection>
          <serverName>JDBCRIPSAadapter</serverName>
          <login>GEO_READ</login>
          <password>GEO_READ</password>
        </serverConnection>
        <queryData>
          <geoFeatureType>RIPS@Topographische
Karten@ÜK1000</geoFeatureType>
        </queryData>
      </serverQuery>
    </queryType>
  </dataSource>
</layer>
<layer enabled="true" editable="false">
  <layerName>ÜK 500</layerName>
  <scale>
    <maxScale>400000.0</maxScale>
    <minScale>175000.0</minScale>
  </scale>
  <dataSource>
    <queryType activeLayer="true">
      <serverQuery>
        <serverConnection>
          <serverName>JDBCRIPSAadapter</serverName>
          <login>GEO_READ</login>
          <password>GEO_READ</password>
        </serverConnection>
        <queryData>
          <geoFeatureType>RIPS@Topographische
Karten@ÜK500</geoFeatureType>
        </queryData>
      </serverQuery>
    </queryType>
  </dataSource>
</layer>
```

(Zu BaWue.mml)

```
<layer enabled="true" editable="false">
  <layerName>TÜK 200</layerName>
  <scale>
    <maxScale>175000.0</maxScale>
    <minScale>80000.0</minScale>
  </scale>
  <dataSource>
    <queryType activeLayer="true">
      <serverQuery>
        <serverConnection>
          <serverName>JDBCRIPSAadapter</serverName>
          <login>GEO_READ</login>
          <password>GEO_READ</password>
        </serverConnection>
        <queryData>
          <geoFeatureType>RIPS@Topographische
Karten@TÜK200</geoFeatureType>
        </queryData>
      </serverQuery>
    </queryType>
  </dataSource>
</layer>
<layer enabled="true" editable="false">
  <layerName>TK 100</layerName>
  <scale>
    <maxScale>80000.0</maxScale>
    <minScale>40000.0</minScale>
  </scale>
  <dataSource>
    <queryType activeLayer="true">
      <serverQuery>
        <serverConnection>
          <serverName>JDBCRIPSAadapter</serverName>
          <login>GEO_READ</login>
          <password>GEO_READ</password>
        </serverConnection>
        <queryData>
          <geoFeatureType>RIPS@Topographische
Karten@TK100</geoFeatureType>
        </queryData>
      </serverQuery>
    </queryType>
  </dataSource>
</layer>
```

(Zu BaWue.mml)

```
<layer enabled="true" editable="false">
  <layerName>TK 50</layerName>
  <scale>
    <maxScale>40000.0</maxScale>
    <minScale>20000.0</minScale>
  </scale>
  <dataSource>
    <queryType activeLayer="true">
      <serverQuery>
        <serverConnection>
          <serverName>JDBCRIPSAdapter</serverName>
          <login>GEO_READ</login>
          <password>GEO_READ</password>
        </serverConnection>
        <queryData>
          <geoFeatureType>RIPS@Topographische
Karten@TK50</geoFeatureType>
        </queryData>
      </serverQuery>
    </queryType>
  </dataSource>
</layer>
<layer enabled="true" editable="false">
  <layerName>TK 25</layerName>
  <scale>
    <maxScale>20000.0</maxScale>
    <minScale>3000.0</minScale>
  </scale>
  <dataSource>
    <queryType activeLayer="true">
      <serverQuery>
        <serverConnection>
          <serverName>JDBCRIPSAdapter</serverName>
          <login>GEO_READ</login>
          <password>GEO_READ</password>
        </serverConnection>
        <queryData>
          <geoFeatureType>RIPS@Topographische
Karten@TK25</geoFeatureType>
        </queryData>
      </serverQuery>
    </queryType>
  </dataSource>
</layer>
</mapModel>
</map>
```

Intranetzusammenstellung der Landschaftsschutzgebiete: UdkLsg.mml

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE map PUBLIC "-//disy Information Systems//MapMarkupLanguage//EN"
'file:/C:/UIS/Dienste/GISterm/etc/MML.dtd'>

<map>
  <version>1.0</version>
  <mapModel>
    <boundingBox>
      <upperLeftX>3388630.0</upperLeftX>
      <upperLeftY>5517091.0</upperLeftY>
      <lowerRightX>3606448.0</lowerRightX>
      <lowerRightY>5264371.5</lowerRightY>
    </boundingBox>
    <layer enabled="true" editable="false">
      <layerName>lsg25_10</layerName>
      <scale>
        <maxScale>75000.0</maxScale>
        <minScale>null</minScale>
      </scale>
      <label layerIsLabeled="false" labelIsScaled="true">
        <labelAttr>FFC</labelAttr>
        <labelAnchor>center</labelAnchor>
        <labelRefScale>1200000.0</labelRefScale>
        <fontSize>11</fontSize>
        <fontName>sansserif</fontName>
        <fontStyle>0</fontStyle>
        <tooltipAttr>none</tooltipAttr>
      </label>
      <dataSource>
        <queryType activeLayer="false">
          <fileQuery>
            <fileType>shapeFileQuery</fileType>
          </fileQuery>
        </queryType>
      </dataSource>
    </layer>
  </mapModel>
</map>
```


(Zu UdkLsg.mml)

```
<URL>[0]lfu/natur_landschaft/Tanja/Internet/Internetdaten/lsg25_10</URL>
  </fileQuery>
</queryType>
</dataSource>
<legend type="SingleSymbol" attribute="" name="lsg25_10"
description="" classSorting="Unsorted">
  <legendClass name="lsg25_10">
    <color red="60" green="150" blue="100" alpha="191" />
    <symbols>
      <fillSymbol hasOutline="false">
        <simpleFillSymbol style="Solid" />
      </fillSymbol>
      <lineSymbol width="1">
        <simpleLineSymbol style="Solid" />
      </lineSymbol>
      <markerSymbol size="10">
        <simpleMarker style="Circle" />
        <color red="60" green="150" blue="100" alpha="191" />
      </markerSymbol>
    </symbols>
  </legendClass>
<undefClass name="unklassifiziert">
  <color red="107" green="124" blue="252" alpha="255" />
  <symbols>
    <fillSymbol hasOutline="true">
      <simpleFillSymbol style="Solid" />
      <outline>
        <color red="0" green="0" blue="0" alpha="255" />
        <lineSymbol width="1">
          <simpleLineSymbol style="Solid" />
        </lineSymbol>
      </outline>
    </fillSymbol>
  </symbols>
</undefClass>
</legend>
</layer>
<layer enabled="true" editable="false">
  <layerName>lsg25_50</layerName>
  <scale>
    <maxScale>250000.0</maxScale>
    <minScale>75000.0</minScale>
  </scale>
  <label layerIsLabeled="false" labelIsScaled="true">
    <labelAttr>FFC</labelAttr>
    <labelAnchor>center</labelAnchor>
    <labelRefScale>1200000.0</labelRefScale>
    <fontSize>11</fontSize>
    <fontName>sansserif</fontName>
    <fontStyle>0</fontStyle>
    <tooltipAttr>none</tooltipAttr>
  </label>
  <dataSource>
    <queryType activeLayer="false">
      <fileQuery>
        <fileType>shapeFileQuery</fileType>
```

(Zu UdkLsg.mml)

```
<URL>[0]lfu/natur_landschaft/Tanja/Internet/Internetdaten/lsg25_50</URL>
  </fileQuery>
</queryType>
</dataSource>
<legend type="SingleSymbol" attribute="" name="lsg25_50"
description="" classSorting="Unsorted">
  <legendClass name="lsg25_50">
    <color red="60" green="150" blue="100" alpha="191" />
    <symbols>
      <fillSymbol hasOutline="false">
        <simpleFillSymbol style="Solid" />
      </fillSymbol>
      <lineSymbol width="1">
        <simpleLineSymbol style="Solid" />
      </lineSymbol>
      <markerSymbol size="10">
        <simpleMarker style="Circle" />
        <color red="60" green="150" blue="100" alpha="191" />
      </markerSymbol>
    </symbols>
  </legendClass>
<undefClass name="unklassifiziert">
  <color red="167" green="26" blue="239" alpha="255" />
  <symbols>
    <fillSymbol hasOutline="true">
      <simpleFillSymbol style="Solid" />
      <outline>
        <color red="0" green="0" blue="0" alpha="255" />
        <lineSymbol width="1">
          <simpleLineSymbol style="Solid" />
        </lineSymbol>
      </outline>
    </fillSymbol>
  </symbols>
</undefClass>
</legend>
</layer>
<layer enabled="true" editable="false">
  <layerName>lsg25_100</layerName>
  <scale>
    <maxScale>750000.0</maxScale>
    <minScale>250000.0</minScale>
  </scale>
  <label layerIsLabeled="false" labelIsScaled="true">
    <labelAttr>FFC</labelAttr>
    <labelAnchor>center</labelAnchor>
    <labelRefScale>1200000.0</labelRefScale>
    <fontSize>11</fontSize>
    <fontName>sansserif</fontName>
    <fontStyle>0</fontStyle>
    <tooltipAttr>none</tooltipAttr>
  </label>
  <dataSource>
    <queryType activeLayer="false">
      <fileQuery>
        <fileType>shapeFileQuery</fileType>
```

(Zu UdkLsg.mml)

```
<URL>[0]lfu/natur_landschaft/Tanja/Internet/Internetdaten/lsg25_100</URL>
  </fileQuery>
</queryType>
</dataSource>
<legend type="SingleSymbol" attribute="" name="lsg25_100"
description="" classSorting="Unsorted">
  <legendClass name="lsg25_100">
    <color red="60" green="150" blue="100" alpha="191" />
    <symbols>
      <fillSymbol hasOutline="false">
        <simpleFillSymbol style="Solid" />
      </fillSymbol>
      <lineSymbol width="1">
        <simpleLineSymbol style="Solid" />
      </lineSymbol>
      <markerSymbol size="10">
        <simpleMarker style="Circle" />
        <color red="60" green="150" blue="100" alpha="191" />
      </markerSymbol>
    </symbols>
  </legendClass>
<undefClass name="unklassifiziert">
  <color red="66" green="224" blue="186" alpha="255" />
  <symbols>
    <fillSymbol hasOutline="true">
      <simpleFillSymbol style="Solid" />
      <outline>
        <color red="0" green="0" blue="0" alpha="255" />
        <lineSymbol width="1">
          <simpleLineSymbol style="Solid" />
        </lineSymbol>
      </outline>
    </fillSymbol>
  </symbols>
</undefClass>
</legend>
</layer>
<layer enabled="true" editable="false">
  <layerName>lsg25_200</layerName>
  <scale>
    <maxScale>1000000.0</maxScale>
    <minScale>750000.0</minScale>
  </scale>
  <label layerIsLabeled="false" labelIsScaled="true">
    <labelAttr>FFC</labelAttr>
    <labelAnchor>center</labelAnchor>
    <labelRefScale>1200000.0</labelRefScale>
    <fontSize>11</fontSize>
    <fontName>sansserif</fontName>
    <fontStyle>0</fontStyle>
    <tooltipAttr>none</tooltipAttr>
  </label>
  <dataSource>
    <queryType activeLayer="false">
      <fileQuery>
        <fileType>shapeFileQuery</fileType>
      </fileQuery>
    </queryType>
  </dataSource>
</layer>
```

(Zu UdkLsg.mml)

```
<URL>[0]lfu/natur_landschaft/Tanja/Internet/Internetdaten/lsg25_200</URL>
  </fileQuery>
</queryType>
</dataSource>
<legend type="SingleSymbol" attribute="" name="lsg25_200"
description="" classSorting="Unsorted">
  <legendClass name="lsg25_200">
    <color red="60" green="150" blue="100" alpha="191" />
    <symbols>
      <fillSymbol hasOutline="false">
        <simpleFillSymbol style="Solid" />
      </fillSymbol>
      <lineSymbol width="1">
        <simpleLineSymbol style="Solid" />
      </lineSymbol>
      <markerSymbol size="10">
        <simpleMarker style="Circle" />
        <color red="60" green="150" blue="100" alpha="191" />
      </markerSymbol>
    </symbols>
  </legendClass>
<undefClass name="unklassifiziert">
  <color red="55" green="133" blue="208" alpha="255" />
  <symbols>
    <fillSymbol hasOutline="true">
      <simpleFillSymbol style="Solid" />
      <outline>
        <color red="0" green="0" blue="0" alpha="255" />
        <lineSymbol width="1">
          <simpleLineSymbol style="Solid" />
        </lineSymbol>
      </outline>
    </fillSymbol>
  </symbols>
</undefClass>
</legend>
</layer>
<layer enabled="true" editable="false">
  <layerName>lsg25_300</layerName>
  <scale>
    <maxScale>1500000.0</maxScale>
    <minScale>1000000.0</minScale>
  </scale>
  <label layerIsLabeled="false" labelIsScaled="true">
    <labelAttr>FFC</labelAttr>
    <labelAnchor>center</labelAnchor>
    <labelRefScale>1200000.0</labelRefScale>
    <fontSize>11</fontSize>
    <fontName>sansserif</fontName>
    <fontStyle>0</fontStyle>
    <tooltipAttr>none</tooltipAttr>
  </label>
  <dataSource>
    <queryType activeLayer="false">
      <fileQuery>
        <fileType>shapeFileQuery</fileType>
      </fileQuery>
    </queryType>
  </dataSource>
</layer>
```

(Zu UdkLsg.mml)

```
<URL>[0]lfu/natur_landschaft/Tanja/Internet/Internetdaten/lsg25_300</URL>
  </fileQuery>
</queryType>
</dataSource>
<legend type="SingleSymbol" attribute="" name="lsg25_300"
description="" classSorting="Unsorted">
  <legendClass name="lsg25_300">
    <color red="60" green="150" blue="100" alpha="191" />
    <symbols>
      <fillSymbol hasOutline="false">
        <simpleFillSymbol style="Solid" />
      </fillSymbol>
      <lineSymbol width="1">
        <simpleLineSymbol style="Solid" />
      </lineSymbol>
      <markerSymbol size="10">
        <simpleMarker style="Circle" />
        <color red="60" green="150" blue="100" alpha="191" />
      </markerSymbol>
    </symbols>
  </legendClass>
<undefClass name="unklassifiziert">
  <color red="199" green="200" blue="247" alpha="255" />
  <symbols>
    <fillSymbol hasOutline="true">
      <simpleFillSymbol style="Solid" />
      <outline>
        <color red="0" green="0" blue="0" alpha="255" />
        <lineSymbol width="1">
          <simpleLineSymbol style="Solid" />
        </lineSymbol>
      </outline>
    </fillSymbol>
  </symbols>
</undefClass>
</legend>
</layer>
<layer enabled="true" editable="false">
  <layerName>lsg25_500</layerName>
  <scale>
    <maxScale>2500000.0</maxScale>
    <minScale>1500000.0</minScale>
  </scale>
  <label layerIsLabeled="false" labelIsScaled="true">
    <labelAttr>NR_ALT</labelAttr>
    <labelAnchor>center</labelAnchor>
    <labelRefScale>1200000.0</labelRefScale>
    <fontSize>11</fontSize>
    <fontName>sansserif</fontName>
    <fontStyle>0</fontStyle>
    <tooltipAttr>NR_ALT</tooltipAttr>
  </label>
  <dataSource>
    <queryType activeLayer="false">
      <fileQuery>
        <fileType>shapeFileQuery</fileType>
```

(Zu UdkLsg.mml)

```
<URL>[0]lfu/natur_landschaft/Tanja/Internet/Internetdaten/lsg25_500</URL>
  </fileQuery>
</queryType>
</dataSource>
<legend type="SingleSymbol" attribute="" name="lsg25_500"
description="" classSorting="Unsorted">
  <legendClass name="lsg25_500">
    <color red="60" green="150" blue="100" alpha="191" />
    <symbols>
      <fillSymbol hasOutline="false">
        <simpleFillSymbol style="Solid" />
      </fillSymbol>
      <lineSymbol width="1">
        <simpleLineSymbol style="Solid" />
      </lineSymbol>
      <markerSymbol size="10">
        <simpleMarker style="Circle" />
        <color red="60" green="150" blue="100" alpha="191" />
      </markerSymbol>
    </symbols>
  </legendClass>
<undefClass name="unklassifiziert">
  <color red="216" green="203" blue="164" alpha="255" />
  <symbols>
    <fillSymbol hasOutline="true">
      <simpleFillSymbol style="Solid" />
      <outline>
        <color red="0" green="0" blue="0" alpha="255" />
        <lineSymbol width="1">
          <simpleLineSymbol style="Solid" />
        </lineSymbol>
      </outline>
    </fillSymbol>
  </symbols>
</undefClass>
</legend>
</layer>
<layer enabled="true" editable="false">
  <layerName>lsg25_1000</layerName>
  <scale>
    <maxScale>3000000.0</maxScale>
    <minScale>2500000.0</minScale>
  </scale>
  <label layerIsLabeled="false" labelIsScaled="true">
    <labelAttr>NR_ALT</labelAttr>
    <labelAnchor>center</labelAnchor>
    <labelRefScale>1200000.0</labelRefScale>
    <fontSize>11</fontSize>
    <fontName>sansserif</fontName>
    <fontStyle>0</fontStyle>
    <tooltipAttr>NR_ALT</tooltipAttr>
  </label>
  <dataSource>
    <queryType activeLayer="false">
      <fileQuery>
        <fileType>shapeFileQuery</fileType>
```

(Zu UdkLsg.mml)

```
<URL>[0]lfu/natur_landschaft/Tanja/Internet/Internetdaten/lsg25_1000</URL>
  </fileQuery>
  </queryType>
</dataSource>
<legend type="SingleSymbol" attribute="" name="lsg25_1000"
description="" classSorting="Unsorted">
  <legendClass name="lsg25_1000">
    <color red="60" green="150" blue="100" alpha="191" />
    <symbols>
      <fillSymbol hasOutline="false">
        <simpleFillSymbol style="Solid" />
      </fillSymbol>
      <lineSymbol width="1">
        <simpleLineSymbol style="Solid" />
      </lineSymbol>
      <markerSymbol size="10">
        <simpleMarker style="Circle" />
        <color red="60" green="150" blue="100" alpha="191" />
      </markerSymbol>
    </symbols>
  </legendClass>
<undefClass name="unklassifiziert">
  <color red="252" green="245" blue="252" alpha="255" />
  <symbols>
    <fillSymbol hasOutline="true">
      <simpleFillSymbol style="Solid" />
      <outline>
        <color red="0" green="0" blue="0" alpha="255" />
        <lineSymbol width="1">
          <simpleLineSymbol style="Solid" />
        </lineSymbol>
      </outline>
    </fillSymbol>
  </symbols>
</undefClass>
</legend>
</layer>
</mapModel>
</map>
```