

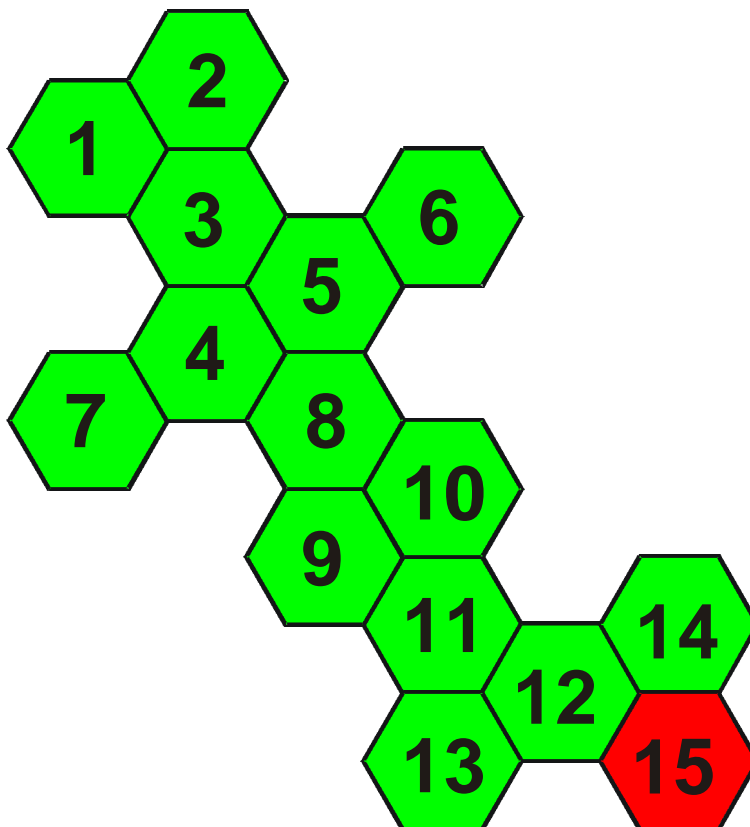


Universitätslehrgang
"Geographical Information Science
& Systems"

UNIGIS MAS 2000

MODUL 15
Projektarbeit:
Straßeninformationssystem
der Stadt Steyr SIS

Horst
Baumgartner



Inhaltsverzeichnis

1. Zusammenfassung	5
2. Problemstellung	6
2.1. Einführung	6
2.2. Probleme beim Aufbau der Datenbasis	6
2.2.1. Bestehende Standards und Richtlinien	6
2.2.2. Definitionsprobleme	6
2.2.3. Probleme bei der Aktualisierung	7
2.2.4. Probleme bei der räumlichen Orientierung	8
2.2.5. Sonstige Probleme	10
3. Anforderungen an das Straßeninformationssystem Steyr (SIS).....	11
3.1. Gesetzliche Rahmenbedingungen	11
3.2. Anforderungen aus der Bauverwaltung	12
4. Projektinformationen	12
4.1. Projektablauf	12
4.2. Projektressourcen	13
4.3. Angaben zum Projektgebiet	13
5. Der Straßengraph der Stadt Steyr (Basisbezugssystem).....	15
5.1. Datenqualität	16
5.1.1. Herkunft der Daten.....	16
5.1.2. Genauigkeit der Beschreibung der räumlichen Lage.....	16
5.1.3. Beschreibung der wichtigsten Attributwerte	19
5.1.4. Richtlinien für die Erfassung und Fortführung	20
5.1.5. Berücksichtigung der zeitlichen Komponente.....	21
5.1.6. Logische Konsistenz des Modells	22
5.2. Datenmodell des Straßengraphen.....	23
5.2.1. Kanten	23
5.2.2. Knoten	24
5.2.3. Straßenverzeichnis	25
5.2.4. Abbiegeregeln (turntables)	25
5.3. Straßenverzeichnis und Plandarstellung	27
6. Datenmodell für die dynamische Segmentierung.....	29
6.1. Bezugspunkte	31
6.2. Routen.....	32
7. Befehlsumfang der SIS-Erweiterung	33
7.1. SIS-Funktionen in der Menüleiste.....	33
7.2. SIS-Funktionen der Buttonleiste	36
7.3. SIS-Funktionen der Werkzeugleiste.....	38
8. Die Erfassung des Busliniennetzes der Stadt Steyr	40
8.1. Methodik beim Aufbau der Datenbasis	40
8.2. Kartographische Aufbereitung der Netzplandarstellung	40
9. Projekttrückschau und Ausblick	43
10. Literatur	44

Anhang A: Straßenkarte Steyr

Anhang B: Avenuescripts

1. Zusammenfassung

Die räumliche Abbildung (Verortung) von Sachverhalten bedingt grundsätzlich ein Raumbezugssystem. Die Verortung von straßenbezogenen Sachverhalten geschieht i.d.R. über Zuordnung zu einer Straße sowie Angabe eines Distanzwertes ab einem bestimmten Bezugspunkt der Straße (d.h. indirekter Raumbezug). Koordinatensysteme als direkte Form eines Raumbezugssystems erscheinen hier nicht problemadäquat (PONN 1996¹).

Im Rahmen der vorliegenden Projektarbeit wurde das geometrische Netzwerk der Straßenachsen der Stadt Steyr durch die Erweiterung mit transparenten topologischen Informationen in ein sogenanntes Betriebsnetz (Straßengraph) überführt. Neben Basisinformationen, die aufgrund eines gesetzlichen Auftrages zur Dokumentation des städtischen Straßennetzes vorzuhalten sind, enthält das Betriebsnetz in eingeschränktem Umfang auch Verkehrsdaten, soweit diese zur Routensuche und –optimierung mit dem Standard-GIS-Produkt ArcView Network Analyst benötigt werden. Neben der Funktion, als Träger begrenzter Informationen, ist das Betriebsnetz auch Bezugssystem für Stationierungsrouten und punktuelle Ereignisse. Stationierungsrouten bauen als Ordnungssystem auf dem Betriebsnetz auf und können einerseits wieder als Basis für stationierte Informationen über mehrere Abschnitte hinweg (Verwaltungsnetz), oder aber auch direkt zur Abbildung linearer Ereignisse auf dem Bezugssystem dienen. Mit dieser Art der Modellierung werden die Rahmenbedingungen für die standardisierte Abbildung und den genormten Austausch von Straßeninformationen geschaffen.

Da die Standardfunktionalität des verwendeten Desktop-GIS (ArcView, Version 3.2) zwar die Anzeige stationierter Informationen auf Basis von Tabellen ermöglicht, nicht aber deren Erfassung, werden im Rahmen des Projektes Werkzeuge entwickelt, mit denen diese Arbeiten unterstützt werden können. Die in Form einer Erweiterung für ArcView vorliegende Programmsammlung enthält außerdem noch Befehle für die Fortführung und Prüfung des Straßengraphen.

Die abschließend durchgeführte Erfassung des städtischen Busliniennetzes von Steyr dient schließlich als praktisches Beispiel für den Einsatz der Werkzeuge beim Aufbau eines Verwaltungsnetzes.

¹ PONN, A. (1996): Salzburger Straßenverkehrsinformationssystem. In: DOLLINGER, F. und J. STROBL: Angewandte Geograph. Informationsverarbeitung VIII = Salzburger Geographische Materialien, Heft 24.

2. Problemstellung

2.1. Einführung

Die Anforderungen an Straßendatenbanken unterscheiden sich teilweise sehr stark voneinander. Bei Straßenerhaltern, die großräumig ausgedehnte Anlagen zu betreuen haben, wurde bereits früh begonnen Daten strukturiert zu erfassen und EDV-technisch zu verarbeiten.

Durch das Aufkommen von kostengünstigen GI-Systemen wird die EDV-gestützte Verwaltung von Straßendaten aber auch für Städte und Gemeinden mittlerer Größe interessant. Gerade die Möglichkeiten die ein GIS bei der Vorhaltung, Visualisierung und Analyse von geographischen Informationen bietet, legen dessen Verwendung beim Aufbau eines Straßeninformationssystems nahe.

Die Situation wird hier am Beispiel der Stadt Steyr beschrieben. Es ist aber anzunehmen, dass die Problematik in Städten mit vergleichbarer Struktur und Größe ähnlich gelagert ist.

2.2. Probleme beim Aufbau der Datenbasis

2.2.1. Bestehende Standards und Richtlinien

Aus der großen Anzahl bestehender Vorschriften und Normen werden hier nur jene aufgelistet, die im Rahmen der Projektarbeit Anwendung oder Erwähnung finden:

- Verordnung der Oö.Landesregierung vom 18.April 1994 über den Inhalt und die Form der Verzeichnisse von Verkehrsflächen der Gemeinde, LGBl. Nr. 41/1994
- RVS: 5.011 Verkehrsinformationssysteme – Grundlagen, Teil A: Bezugssystem für Straßen
- RVS 5.021 Stationierung von Straßen - Grundlagen der Stationierung
- RVS 5.022 Stationierung von Straßen - Kilometerzeichen
- ArcView Network Analyst (*setting up a network*)

2.2.2. Definitionsprobleme

Ein Vergleich der Straßengattungen nach der RVS und der landesgesetzlichen Regelung für Oberösterreich lässt bereits erkennen, dass die Kombination der beiden Richtlinien nur begrenzt realisierbar ist.

Straßengattungen lt. Pkt 4. Schlüssel für Straßen RVS 5.011 Verkehrsinformationssysteme –Grundlagen Bezugssystem für Straßen

1. Bundesstraßenkategorien
 - 1.1. A-Autobahnen
 - 1.2. B-Bundesstraßen
 - 1.3. S-Schnellstraßen
 - 1.4. P-Privatstraßen des Bundes
2. Landesstraßenkategorien
 - 2.1. L-Landesstraßen
 - 2.2. K-Konkurrenzstraßen

- 2.3. E-Eisenbahnzufahrtsstraßen
- 2.4. F-Sonstige Straßen
- 3. Sonstige Straßen
 - 3.1. H-Gemeindestraße 1.Ordnung Hauptstraße
 - 3.2. G-Gemeindestraße 2. Ordnung, Sammelstraße oder Gemeindestraße ohne hierarchische Gliederung
 - 3.3. Z-Gemeindestraße 3.Ordnung, Erschließungsstraße
 - 3.4. V-Privatstraße, Genossenschaftsstraße, Gesellschaftsstraße
 - 3.5. W-Sonstiger Weg, wie öffentlicher Interessentenweg, Güterweg, Forstweg
 - 3.6. O-Straßennummer (Gemeindestraße) nach dem Straßenverzeichnis des ÖSTAT
- 4. Ergänzungsstraßen
 - 4.1. E-Einbahn in Richtung der Kilometrierung
 - 4.2. G-Einbahn gegen die Richtung der Kilometrierung
 - 4.3. K-Kreisverkehr
 - 4.4. A-Straßenast
 - 4.5. N-Nebenfahrbahn
 - 4.6. R-Rampe
 - 4.7. B-Betriebsweg
 - 4.8. Z-Zu-und Abfahrt zu einem Park- bzw. Rastplatz etc.
 - 4.9. P-Park-,Rastplatz
 - 4.10. V-Verbindungsstraße
 - 4.11. S-Sonstige Ergänzungsstraße

Straßengattungen nach § 8 Abs. 2 O.ö. Straßengesetz 1991 (LGBl. 82/1997)

- 1. Landesstraßen
- 2. Verkehrsflächen der Gemeinde:
 - 2.1. Gemeindestraßen
 - 2.2. Güterwege
 - 2.3. Radfahrwege, Fußgängerwege und Wanderwege
- 3. Ortschaftswege

2.2.3. Probleme bei der Aktualisierung

Die Verortung von Informationen über den Straßenraum, bedingt grundsätzlich ein räumliches Bezugssystem, in dem jeder Punkt im Straßennetz durch Angabe der Straßenabschnittsnummer und dem Abstand vom Beginn der Kante eindeutig definiert ist. Für Verkehrsnetze haben sich Bezugssysteme bewährt, die den Raumbezug von Stationierungsrouten abzuleiten, die ihrerseits bereits auf eine Basisgeometrie aufbauen. Problematisch ist in diesem Zusammenhang die Fehleranfälligkeit des Systems bei Veränderungen der Basisgeometrie. Werden die Straßenzüge, beispielsweise nach Straßennamen über mehrere Kanten hinweg, durchgehend kilometriert, führt jede geometrische Veränderung automatisch zu einer Fehlstationierung. Informationen, die auf das ursprüngliche Bezugssystem aufbauen, können dadurch in ihrer Lage beeinflusst werden.

Abhilfe kann nur dann geschaffen werden, wenn eine solcherart betroffene Kante des Bezugssystems nicht gelöscht, sondern lediglich als ungültig gekennzeichnet wird. Diese Kennzeichnung erfolgt sinnvoller Weise mit dem Datum der Deaktivierung der Kante. Informationen, die sich auf ungültige Kanten beziehen, bleiben auf diese Weise in ihrer ursprünglichen Form rekonstruier- und darstellbar.

Die Festlegung einer von der Straßenbezeichnung abgeleiteten Bezugslinie stellt ein weiteres Problem dar. Welche Schwierigkeiten sich aufgrund der heterogenen Struktur dabei in manchen Gebieten ergeben können, lässt sich anhand der nachfolgenden Abbildung der Achsen der Straßenanlage "Wieserfeldplatz" in Steyr erahnen:

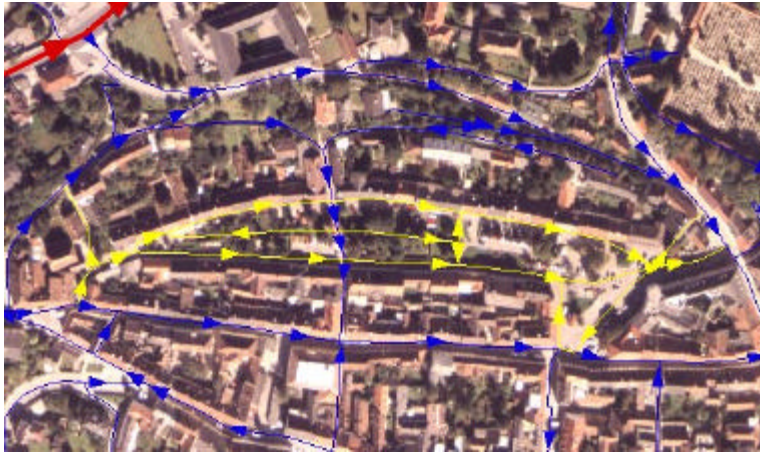


Abbildung 1: Straßenanlage Wieserfeldplatz (gelb dargestellt)

2.2.4. Probleme bei der räumlichen Orientierung

Zu den Schwierigkeiten die sich bei der Fortführung des Straßengraphen ergeben, kommen noch jene, die bei der räumlichen Orientierung in einem Straßennetz auftreten können.

Widrigkeiten, mit denen man beispielsweise bei der Verortung von Unfallgeschehen konfrontiert ist, wurden bereits ausführlich von MAURER² beschrieben.

In einem Beitrag von GOBIET³ zur RVS 5.011 wird ein Bezugssystem für Verkehrs- und Verkehrswegedatenbanken in Österreich vorgestellt. Im Rahmen der Projektarbeit wurde die Richtlinie auf ihre zukünftige Anwendbarkeit hin untersucht.

Grundelemente der Verortung von Ereignissen nach der RVS 5.011 sind in der Natur vorhandene markante Punkte (Kilometerzeichen, Notrufsäulen,...). Gemeindestraßen werden in der Natur üblicherweise nicht mit Kilometerzeichen stationiert. Eine Übertragung der Stationierung in die Natur erscheint aufgrund der kurzen Straßenlängen auch nicht sinnvoll. Gerade die Festlegung solcher Bezugspunkte stellt nun aber das größte Problem beim Versuch der Anwendung der RVS im innerstädtischen Bereich dar. Die Verwendung der Orientierungsnummer eines Bauwerkes als Bezugspunkt ist aufgrund der damit verbundenen Unschärfe (Hausabmessungen) für bestimmte Themen ungeeignet. Werden Straßeneinbauten, wie Wasserleitungsschieber oder Kanalschächte als Bezugspunkte herangezogen, besteht in vielen Fällen die Gefahr einer Verwechslung. Auch der Vorschlag, im untergeordneten Straßennetz Straßenkreuzungen als eindeutige Fixpunkte zu verwenden und die Bezeichnung dieser Punkte aus den Namen der

² MAURER, P: Visualisierung des Unfallgeschehens in einem Geographischen Informationssystem (GIS). . In: ZAGEL, B. (Hrsg.): GIS in Verkehr und Transport, Heidelberg 2000

³ GOBIET, W: Datenmodell für ein einheitliches Bezugssystem für Verkehrs- und Verkehrswegedatenbanken in Österreich. In: ZAGEL, B. (Hrsg.): GIS in Verkehr und Transport, Heidelberg 2000

kreuzenden Verkehrsflächen abzuleiten, führt nicht unbedingt zu eindeutigen Bezugspunkten, wie die nachfolgende Skizze zeigt.

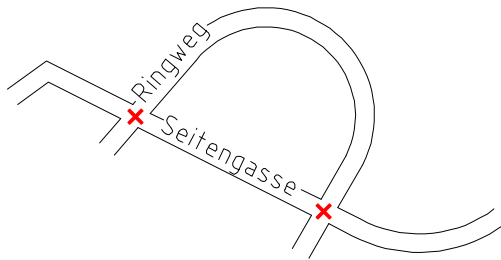


Abbildung 2: Doppelkreuzung

Der Austausch und die Darstellung von Informationen auf Basis von Bezugspunkten nach den Richtlinien der RVS 5.011, setzt im Ziel- und im Quellsystem zumindest die identische Orientierung der Bezugslinien voraus. Da empfohlen wird, den Bezugspunkt zur Wiederauffindbarkeit in der Natur mit Rechts- und Hochwert zu versehen, könnte die Vernachlässigung der Kantenorientierung (bei ausreichender Genauigkeit) bei der Datenerfassung und beim Austausch von Straßeninformationen zu Vereinfachungen führen. Ereignisse, die bei einer Projektion auf die Straßenachse links vom Bezugspunkt liegen, könnten generell mit einem negativen Abstandswert, rechts liegende mit einem positiven Abstandswert versehen werden. In den nachstehenden vier Abbildungen wurde zuerst die Orientierung nach RVS 5.011 schematisch dargestellt und anschließend jene, die bei Verwendung eines neutralen Bezugspunktesystems, ohne Berücksichtigung der Kantenorientierung, zur Anwendung kämen.

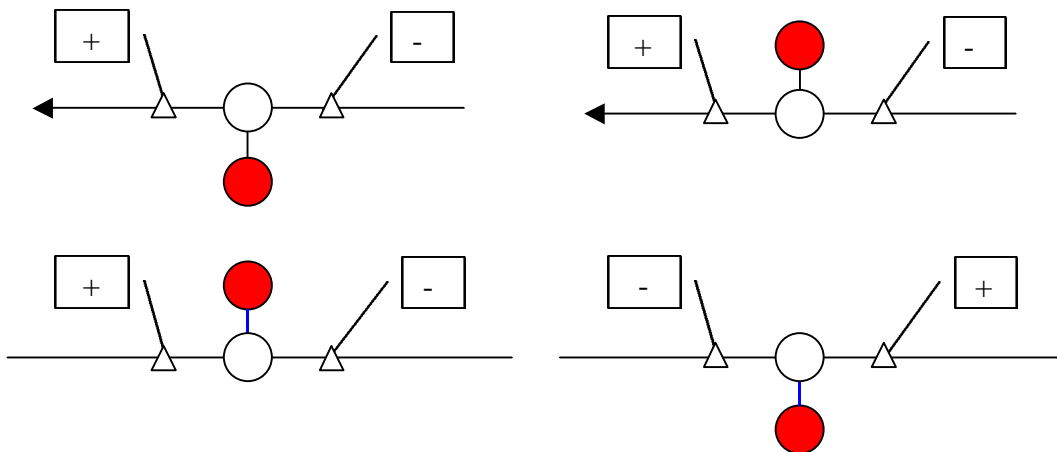


Abbildung 3: Verortung von Ereignissen

Lediglich die Lage des Bezugspunktes in Richtung zur Kante würde die Festlegung der Orientierung auf der Straßenachse ermöglichen. Die Kenntnis der Kantentrichtung wäre für eine Verortung von Ereignissen vom Bezugspunkt aus, so nicht mehr erforderlich. Die zwingende Einhaltung von Standards lässt in diesem Zusammenhang aber momentan keine alternative Lösung zu.

2.2.5. Sonstige Probleme

Zusammengefasst wirken sich noch folgenden Punkte nachteilig beim Aufbau eines Informationssystems für Stadtstraßennetze aus:

- In städtischen Gebieten wurde in vielen Fällen bei der Projektierung von Straßen auf eine koordinative Berechnung der Achse verzichtet.
- In den historischen Teilen der Stadt existieren für die meisten Straßenanlagen keine Projektunterlagen. (Anm.: Im Fall der Stadt Steyr, mit einer über tausend Jahre alten Geschichte, ist dieser Umstand nicht weiter verwunderlich.)
- Die Straßennamen lassen in den meisten Fällen alternative Schreibweisen zu. Dadurch kann es bei der verteilten Sammlung von Informationen zu Problemen kommen. (z.B.: Sierninger Straße, Sierningerstraße, Sierninger-Straße,...)
- Untergeordnete Teile des Straßennetzes sind oftmals überhaupt nicht bezeichnet.
- In manchen Fällen spalten sich Straßenanlagen mit identischer Bezeichnung auf oder enthalten abzweigende Stichstraßen mit dem selben Namen.
- Die Anzahl der Richtlinien auf diesem Gebiet ist momentan unübersichtlich groß. Standards, die beispielsweise für die Fahrzeugnavigation und die Verkehrstelematik entwickelt werden, können kaum im Bereich des Infrastrukturmanagements eingesetzt werden (trotz großteils identer Dateninhalte !)⁴.
- Personelle Ressourcen für die Wartung des Systems sind nur begrenzt vorhanden.

⁴ KOLLARITS S: Verkehr ist mehr. Modellgrundlagen zur (Re-)Integration von Verkehrsinfrastruktur, Verkehrsplanung und Verkehrsmanagement. In: SCHRENK M. (Hrsg.): CORP 99 Beiträge zum 4. Symposium zur Rolle der Informationstechnologie in der Raumplanung, Wien 1999

3. Anforderungen an das Straßeninformationssystem Steyr (SIS)

3.1. Gesetzliche Rahmenbedingungen

In § 9 des O.ö. Straßengesetzes von 1991 (LGBl.Nr. 84/1991) ist der Auftrag zur Führung von Straßenverzeichnissen geregelt⁵:

§ 9

Straßenverzeichnisse

(1) Das Land hat ein Verzeichnis der Verkehrsflächen des Landes zu führen. Jede Gemeinde hat ein Verzeichnis ihrer Verkehrsflächen zu führen.

(2) Die Verzeichnisse sind öffentlich. Es steht jedermann frei, in die Verzeichnisse einzusehen, davon an Ort und Stelle Abschriften selbst anzufertigen oder nach Maßgabe der vorhandenen technischen Möglichkeiten auf seine Kosten Kopien anfertigen zu lassen.

(3) In die Verzeichnisse sind die Verkehrsflächen - gesondert nach Straßengattungen fortlaufend numeriert - mit ihrem Verlauf (Änderungen des Verlaufes) einzutragen; die Verzeichnisse sind mit Straßenkarten zu versehen. Die Landesregierung hat durch Verordnung Inhalt und Form der Verzeichnisse näher zu regeln.

Die nähere Regelung über Inhalt und Form der Straßenverzeichnisse ist in LGBl. 41/1994 definiert und wird hier auszugsweise wiedergegeben:

§ 1

Jede Gemeinde hat über alle innerhalb ihres Gemeindegebietes liegenden öffentlichen Verkehrsflächen der Gemeinde ein Verzeichnis zu führen. Es ist entsprechend den Straßengattungen (§ 8 Abs. 2 O.ö. Straßengesetz 1991) zu gliedern.

§ 2

Die einzelnen Verkehrsflächen der Gemeinde sind innerhalb der Straßengattungen nach ihrem jeweiligen Namen oder ihrer Umschreibung in alphabetischer Reihenfolge aufzulisten und mit einer fünfstelligen Zahl zu versehen. Die erste Ziffer dieser Zahl soll nach Möglichkeit die Straßengattung gemäß § 8 Abs. 2 O.ö. Straßengesetz 1991 ausdrücken. Verkehrsflächen, deren Straßengattung nicht bestimmt ist, sind als Ortschaftswege zu führen.

§ 3

Für jede Verkehrsfläche der Gemeinde soll überdies ein eigenes Datenblatt angelegt werden, in welchem Anfang, Verlauf und Ende der jeweiligen Verkehrsfläche näher zu beschreiben sind.

§ 4

(1) Dem Verzeichnis ist eine Straßenkarte im Maßstab 1:20.000 für das gesamte Gemeindegebiet anzuschließen. In der Straßenkarte sind in jedem Fall neben nennenswerten Orten die allfälligen Bundesstraßen

⁵ <http://www.ris.bka.gv.at/lr-oberoesterreich/>

und Verkehrsflächen des Landes entsprechend der Straßenkarte für Oberösterreich und sämtliche öffentlichen Verkehrsflächen der Gemeinde entsprechend ihrer Gattung und zugeordneten Zahl auszuweisen.

(2) Gemeindestraßen müssen braun, Ortschaftswege gelb, Güterwege orange, Radwege strichpunktiert und Fußgänger- und Wanderwege mit einer schwarzen Linie gekennzeichnet sein.

(3) Erforderlichenfalls kann für dicht besiedelte Gebiete eine gesonderte Straßenkarte in einem größeren Maßstab angelegt werden.

§ 5

Das Straßenverzeichnis einschließlich der Straßenkarte ist laufend zu aktualisieren.

3.2. Anforderungen aus der Bauverwaltung

Neben den Vorschriften des Landesgesetzgebers werden die Anforderungen aus der Bauverwaltung wie folgt definiert:

- Die digitale Abbildung des Straßennetzes von Steyr mit seinen grundlegenden Verkehrsregeln als Knoten- Kanten-Struktur in ArcView-GIS.
- Möglichst transparentes Datenformat für die Nutzung außerhalb der GIS-Umgebung.
- Zugriff auf Netzwerk-Topologieinformationen ohne Erfordernis der Network Analyst Extension von Esri.
- Die Möglichkeit zur Datennutzung bei Vorliegen der Network Analyst Extension ohne Änderung der Struktur.
- Die Verwendbarkeit des Straßengraphen (*Betriebsnetz*) als Bezugssystem zur Abbildung von linearen oder punktförmigen Informationen mit Hilfe der dynamischen Segmentierung.
- Berücksichtigung des Zeitfaktors bei der Datenpflege zur nachvollziehbaren Dokumentation der Veränderungen im Straßennetz.
- Problemlose Mehrfachnutzung der im Rahmen des Projektes entwickelten Werkzeuge durch Bereitstellung als ArcView-Extension (*SIS-Extension*).

4. Projektinformationen

4.1. Projektablauf

- Entwicklung des Datenmodells
- Entwicklung von Werkzeugen für die Abbildung, Prüfung, Fortführung und weitere Nutzung als Referenzsystem
- Digitale Erfassung des Straßennetzes und der Netzregeln
- Kartographische Aufbereitung der Straßenkarte
- Erfassung und Abbildung der Haltestellen und Linienführung der städtischen Verkehrsbetriebe Steyr unter Verwendung des Straßengraphen als Bezugssystem

4.2. Projektressourcen

- Personelle Ressourcen:
Die Bearbeitung erfolgt im Rahmen des UniGIS MAS-Lehrganges 2000.
- Technische Ressourcen:
PC-Pentium 128MB
Betriebssystem Windows 98
- GI-System:
ArcView-Gis V 3.2a mit Network Analyst v1.0b
AutoCad-Map Release 3
- Verwendete Basisdaten:
Digitale Katastralmappe des Bundesamtes für Eich- und Vermessungswesen (DKM)
Grundstücksdatenbank des Bundesamtes für Eich- und Vermessungswesen (GDB)
Orthophotobestand der Stadt Steyr (Stand Herbst 1997)
Radwegekonzept Steyr
Straßenverzeichnis der Fachabteilung für Statistik und Einwohnerwesen
Fahrplan und Netzbeschreibung der Stadtbetriebe Steyr

4.3. Angaben zum Projektgebiet



Abbildung 4: Lage der Stadt Steyr in Oberösterreich

Die oberösterreichische Bezirkshauptstadt Steyr erstreckt sich über eine Fläche von rund 26,6 km². Mit seinen rund 40.000 Einwohnern ist Steyr die drittgrößte Stadt des



Bundeslandes. Das Stadtbild wird stark von den beiden Alpenflüssen Enns und Steyr geprägt. Der gut erhaltene historische Stadtkern liegt direkt an der Mündung der Steyr in die Enns. Die im vorigen Jahrhundert entstandenen Fabriken und Handwerksbetriebe haben sich größtenteils entlang des Steyr-Flusses am Rande der Altstadt angesiedelt, da hier ausreichend Möglichkeit zur Nutzung der Wasserkraft gegeben war. Großbetriebe, die dann später im 20. Jahrhundert in Steyr entstanden sind, mussten auf die unbebauten Konglomeratterrassen entlang der Enns ausweichen. Die zahlreichen neuen Wohnsiedlungen wie Tabor, Ennsleite, Resthof oder Münchenholz haben sich in weiterer Folge um den Stadtkern herum entwickelt.

Abbildung 5: Blick entlang des Steyr-Flusses in Richtung Michaelerkerche

Topographische Basisdaten⁶:

Seehöhe	310 m
Geographische Koordinaten	48° 03' 00" Breite, 14° 25' 00" Länge
Ost-West Ausdehnung	7 km
Nord-Süd Ausdehnung	7,3 km

Flächenverteilung (nach Benützung in km²):

Wald	2,8 km ²
Landwirtschaftliche Nutzung	11,1 km ²
Baufläche	2,1 km ²
Gärten	5,3 km ²
Gewässer	1,6 km ²
Sonstige Flächen	3,7 km ²

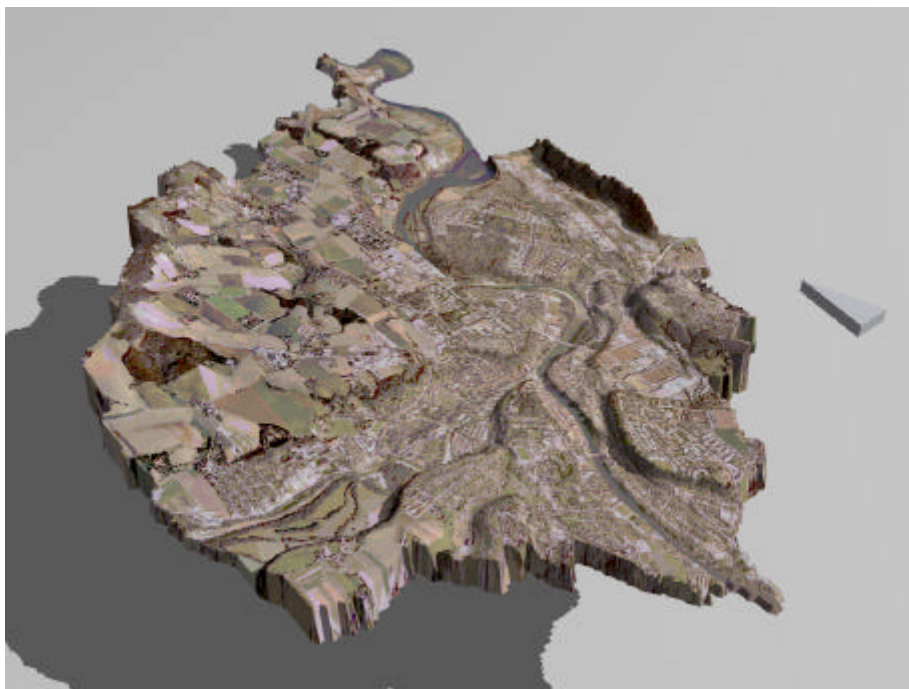
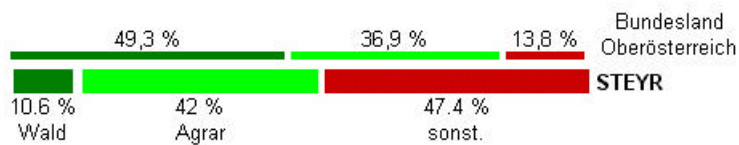


Abbildung 6: Übersichtsdarstellung (DHM der Stadt Steyr mit Orthophototextur, Stand 1997)

⁶ Land Oberösterreich, (2002-01-04), <http://www.ooe.gv.at/geographie/geoinfo/gem/g40201i.htm>

5. Der Straßengraph der Stadt Steyr (Basisbezugssystem)

Ausgehend von der bereits vorliegenden Basisgeometrie wird, durch ergänzende Attributinformation der Kanten, ein Betriebsnetz (*Straßengraph*) aufgebaut.

Das Betriebsnetz hat dabei folgende Funktionen:

- Basisbezugssystem für Verwaltungsnetze
- Träger von Topologieinformationen
- Träger von Informationen für das Straßenverzeichnis nach dem O.ö.Straßengesetz
- Datengrundlage für den Einsatz der Anwendung *Network Analyst* der Firma ESRI.

Zur Untergliederung des Straßengraphen wurde die, im Vergleich zu den Bestimmungen der RVS 5.011, relativ einfache Kategorisierung nach den Landesvorschriften herangezogen und lediglich um die Gattung der Bundesstraßen erweitert. Die Erfüllung des gesetzlichen Auftrages einer oberösterreichischen Gemeinde zur Dokumentation des Straßennetzes kann auf diese Weise bereits mit dem Betriebsnetz in ausreichender Form sichergestellt werden. Die Nutzung der Informationen im Rahmen der Anforderungen aus der Bauverwaltung der Stadt Steyr bleibt dabei ebenfalls gewährleistet.

Komplexere Anwendungen wie Verkehrsinformationssysteme, die bereits in die Gruppe der Verwaltungsnetze einzuordnen sind, bedingen bereits ein auf den Straßengraphen aufbauendes Ordnungssystem in Form von Stationierungsrouten.

Die nachstehende Abbildung von S.KOLLARITS (1999)⁷ beschreibt den Zusammenhang von Betriebs- und Verwaltungsnetzen anhand der Elemente des Datenmodells.

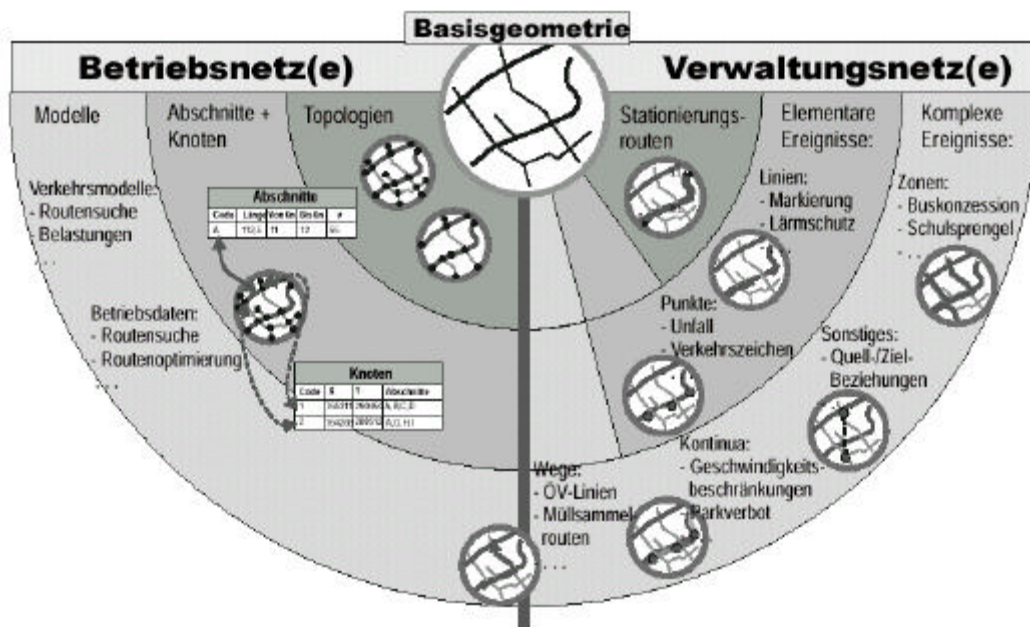


Abbildung 7: Elemente des Datenmodells für Verwaltungs- und Betriebsnetze

Die Abbildung von Stationierungsrouten auf dem Straßengraphen wird im Abschnitt "Datenmodell für die dynamische Segmentierung" näher erläutert.

⁷ KOLLARITS S: Verkehr ist mehr. Modellgrundlagen zur (Re-)Integration von Verkehrsinfrastruktur, Verkehrsplanung und Verkehrsmanagement. In: SCHRENK M. (Hrsg.): CORP 99 Beiträge zum 4. Symposium zur Rolle der Informationstechnologie in der Raumplanung, Wien 1999

5.1. Datenqualität (Metadaten)

Die Datenqualität kann bezüglich verschiedener Aspekte betrachtet und beschrieben werden (GUPTILL und MORRISON 1995⁸):

- Herkunft der Daten
- Genauigkeit der Beschreibung der räumlichen Lage
- Genauigkeit der Attributwerte
- Vollständigkeit des Modells
- Logische Konsistenz des Modells
- Semantische Genauigkeit
- Genauigkeit der zeitlichen Angaben

Die nachfolgende Beschreibung der Datenqualität des Straßengraphen der Stadt Steyr folgt ansatzweise dieser Gliederung:

5.1.1. Herkunft der Daten

Der Grund für die bereits mehrere Jahre zurückliegende Ersterfassung des Straßengraphen von Steyr lag im Erfordernis sogenannte Service-Areas abzuleiten und kürzeste Wege im Straßennetz der Stadt bestimmen zu können.

Die Ersterfassung wurde von der Vermessungsabteilung der Stadt Steyr durchgeführt. Als Grundlage diente die digitale Katastralmappe des Bundesamtes für Eich- und Vermessungswesen (DKM), die für den Raum Steyr aufgrund einer terrestrischen Neuvermessung mit einer relativ hohen Genauigkeit von ± 0.2 m vorliegt. Die erforderlichen Sachdaten über den Grundbesitz stammen aus der Grundstücksdatenbank des Bundesamtes für Eich- und Vermessungswesen und werden von der Stadt Steyr zur Abwicklung der ihr gesetzlich vorgeschriebenen Aufträge in regelmäßigen Abständen angekauft.

Durch die Auswahl aller Grundstücksflächen der öffentlichen Straßenerhalter (getrennt nach Bund, Land, Gemeinde) konnte ein Ausgangsdatenbestand erzeugt werden. Die komfortablen grafischen Konstruktionsmöglichkeiten waren der Grund für die Verwendung von AutoCad-Map für die anschließende Erfassung der Straßenkanten. Zur Unterscheidung der Straßenerhalter wurde dort mit unterschiedlichen AutoCad-Layern gearbeitet. Bei der anschließenden Bildschirmdigitalisierung wurden die Kanten entweder in Mitte der begleitenden Grundgrenzen oder, wenn in der DKM gesondert ausgewiesen, in Mitte der Fahrbahn festgelegt. Für die Attributierung der Kanten mit Sachdaten wurde der Datenbestand wieder in das ArcView-Shapeformat überführt.

5.1.2. Genauigkeit der Beschreibung der räumlichen Lage

Bezugssystem der Achsengeometrie ist das System der Österreichischen Landesvermessung Gauß-Krüger M31°.

Die zum Zeitpunkt der Digitalisierung getroffene Annahme, dass sich die Straßenachse immer in der Mitte der begleitenden Grundgrenzen befindet, ist nur bedingt haltbar. Im Bereich von Überlandstraßen sind auch Straßenböschungen und Geländeeinschnitte in

⁸ GUPTILL, S.C. und MORRISON, J.L., 1995: Elements of spatial data quality. Oxford: Elsevier Science Ltd.

den Straßenparzellen enthalten, wodurch, trotz der relativ hohen Genauigkeit der verwendeten Kartengrundlage, eine grobe Lageüberprüfung nicht nur zur Beschreibung der Daten sinnvoll erscheint.

Die Bestimmung der Lagegenauigkeit des Straßengraphen erfolgte durch eine Zufallsstichprobe aus der Grundgesamtheit von 2036 Kanten.

Mit dem Avenue-Script *SIS.MkRandomPoint* wurden in einem ersten Schritt einhundert Punkte in der Mitte zufällig ausgewählter Kanten generiert. Anschließend waren die Punkte am Bildschirm so zu verschieben, dass sie unter Zuhilfenahme von Orthophoto und Kataster auf der tatsächlichen Straßenachse zu liegen kamen.

Für jene Punkte, deren Position auf der Straßenachse mit dieser Methode, unzweifelhaft bestimmt werden konnte, wurde anschließend der kürzeste Abstand zur nächsten Kante mit dem Skript *SIS.PtToLineTbl* bestimmt. Die Programmsequenz bietet die Möglichkeit Punkte auf die nächstgelegene Kante zu projizieren. Dabei werden die Stationierung zum Anfangspunkt und der Abstand des Punktes von der Kante berechnet. Da nur dieser Abstand des Punktes von der Straßenachse für die weitere Untersuchung von Interesse ist, wirkt sich eine geringfügige Verschiebung entlang der Straßenachse bei einer Neupositionierung nicht nachteilig aus.

Die Verteilung der Lagefehler einer Linie ist nach G. DUTTON⁹ nicht als konstant breiter Korridor zu sehen, sondern ist an den Stützpunkten breiter als zwischen diesen. Daher sind die nachfolgend angegebenen Werte auch nur als grobe Schätzung der Genauigkeit des Straßengraphen zu verstehen. Dies auch deshalb, weil der Interpretationsspielraum des Bearbeiters bei der Neupositionierung der Punkte auf die *korrekten Werte* nur schwer abgeschätzt werden kann.



Abbildung 8: Kombination aus Straßengraph, Orthophoto und DKM

63 von 100 zufällig erzeugten Punkte konnten auf Basis der verwendeten Unterlagen neu positioniert werden. 27 Punkte waren aufgrund einer unzureichenden Katasterdarstellung oder mangelnder Bildqualität auszuschneiden.

⁹ DUTTON, G. (1992): Handling Positional Uncertainty in Spatial Databases. In: Proceedings SHD'92: 460-469

Für die Stichprobe über die Abweichung zwischen Ist- und Soll-Lage können folgende Schätzwerte angegeben werden:

Mittelwert: $\bar{x} = 0,488 \text{ m}$

Varianz: $s_x^2 = 0,617 \text{ m}^2$

Standardabweichung $s_x = 0,785 \text{ m}$

Freiheitsgrad:

$$f = 63 - 1 = 62$$

Bestimmung der Vertrauensgrenzen für den Mittelwert:

Vertrauensniveau $P = 1 - \alpha$

Standardabweichung des Mittelwertes:

$$s_x^- = s_x / \sqrt{n}$$

$$s_x^- = 0,785 / \sqrt{63} = 0,099 \text{ m}$$

Schranken für die t-Verteilung

$$t_{0,975;62} = 1,999$$

$$t_{0,995;62} = 2,658$$

Schätzwerte der unteren und oberen Schranken:

5% Irrtumswahrscheinlichkeit:

$$c_{i,u} = 0,488 - 0,09953 \cdot 1,999 = 0,289 \text{ m}$$

$$c_{i,o} = 0,488 + 0,09953 \cdot 1,999 = 0,687 \text{ m}$$

1% Irrtumswahrscheinlichkeit:

$$c_{i,u} = 0,488 - 0,09953 \cdot 2,658 = 0,223 \text{ m}$$

$$c_{i,o} = 0,488 + 0,09953 \cdot 2,658 = 0,752 \text{ m}$$

Vertrauensgrenzen für die Standardabweichung bei 5% Irrtumswahrscheinlichkeit:

Schranken der χ^2 - Verteilung:

$$\chi^2_{0,025;62} = 42,198$$

$$\chi^2_{0,975;62} = 85,581$$

Damit lassen sich die unteren und oberen Schranken wie folgt abschätzen:

$$c_{\sigma,u} = 0,785 \cdot \sqrt{62/85,581} = 0,668 \text{ m}$$

$$c_{\sigma,o} = 0,785 \cdot \sqrt{62/42,198} = 0,952 \text{ m}$$

Mit 95% Wahrscheinlichkeit liegt die wahre Streuung der Probe im Bereich von 0,67 und 0,95 m.

5.1.3. Beschreibung der wichtigsten Attributwerte

Straßengattung und Alternativbezeichnung:

Die Feststellung des Straßenerhalters und damit der Straßengattung erfolgt auf Basis einer Eigentümererhebung im Kataster (sh. Herkunft der Daten).

Die Straßenkarte des Landes Oberösterreich (Amt der OÖ Landesregierung, Stand Mai 1998) dient in weiterer Folge zur Feststellung der Alternativbezeichnung für Bundes- und Landesstraßen.

Kategorie:

Die Werte im Feld Wegekategorie (sh. dazu Punkt 5.2.1) entsprechen dem Wissensstand des Datenerfassers zum Zeitpunkt der Ersterhebung. Die weitere Fortschreibung erfolgt durch die Einarbeitung straßenrechtlicher Bewilligungen nach dem OÖ Straßengesetz.

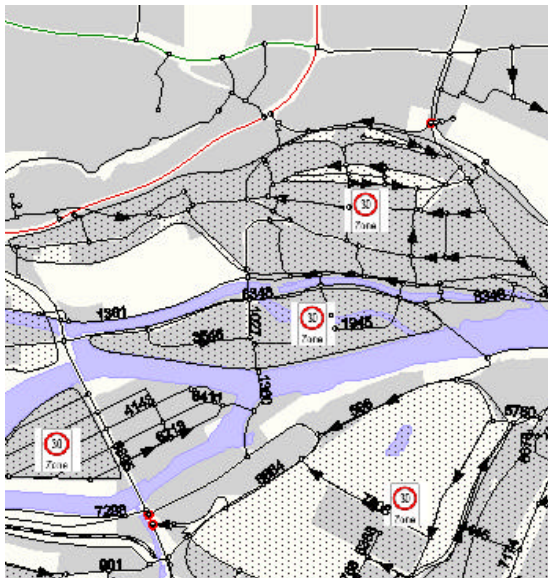
Straßennamen:

Die Grundlage für die Bezeichnung der Straßen bildet das offizielle Straßenverzeichnis der Abteilung Statistik und Einwohnerwesen der Stadt Steyr (Stand Dezember 2001). Die Attributierung der Kanten mit dem Straßenschlüssel erfolgte auf Basis analoger Straßenpläne unter Zuhilfenahme des digitalen Hausnummernplanes der Stadt Steyr. Der Straßenschlüssel 0 beschreibt dabei Kanten ohne Straßennamen. Mit der Schlüsselzahl 9999 wurden Kanten außerhalb des Gemeindegebietes gekennzeichnet. Bei der Ableitung einer Straßenstatistik bleiben letztere unberücksichtigt.

Geschwindigkeitsbegrenzungen, vorgeschriebene Fahrtrichtung und Fahrverbote:

Die zulässigen Höchstgeschwindigkeiten der Kanten wurden in einem ersten Schritt aus dem Kantenattribut 'Kategorie' nach der folgenden Regel abgeleitet:

- Straßen 50 km/h
- Rad-und Gehwege 4 km/h



Für die Nachbearbeitung standen Geodaten aus der Erhebung zum Radkonzept Steyr¹⁰ zur Verfügung, die durch aktuelle Informationen von der Fachabteilung für Stadtentwicklung und Stadtplanung ergänzt wurden.

Jene Kanten, die sich zur Gänze innerhalb der Zonendefinition aus dem Radkonzept befanden, wurden mit dem dort eingetragenen Geschwindigkeitswert versehen. Da die Gebietsdefinitionen seinerzeit ohne Rücksicht auf die Straßenachsen festgelegt wurden, war eine manuelle Nachbearbeitung der Kanten im Grenzbereich der Flächen notwendig.

Abbildung 9: Zonenbeschreibung Radkonzept

Die weiteren Verkehrsdaten über Einbahnen und Abbiegeregeln stammen ebenfalls von der oben genannten Abteilung.

¹⁰ Dipl.Ing. Klaus Girkingner im Auftrag der Stadt Steyr, Stand November 1997

Ein Wechsel der zulässigen Höchstgeschwindigkeiten auf einer Kante ist im vorliegenden Datenmodell kein Grund für einen Attributwechsel, und somit auch nicht für die Unterbrechung der Kante. In solchen Fällen ist jene Geschwindigkeitsbeschränkung die den längsten Bereich der Kante in Anspruch nimmt abzubilden.

Anm.: Sollte sich zeigen, dass die Thematik auf diese Art und Weise nicht ausreichend genau beschrieben werden kann, steht eine Abbildung als eigenes Routenthema offen.

Zeitdauer:

Das Kantenattribut für die Zeitdauer (in Sekunden), die zur Überwindung einer Kante aufzuwenden ist, errechnet sich aus der Länge der Kante und der zulässigen Höchstgeschwindigkeit nach der Formel:

$\text{Sekunden} = (\text{Länge in Metern} * 3.6 / \text{Geschwindigkeit in km/h})$

Vollständigkeit des Modells:

Derzeit enthält das Modell alle Straßen- und Wegeverbindungen im Stadtgebiet von Steyr, soweit sich diese im Eigentum von Bund, Land Oberösterreich oder der Stadt Steyr in ihrer Funktion als Straßenerhalter befinden. Vereinzelt sind auch Verkehrsflächen privater Eigentümer enthalten, wenn diese öffentlich nutzbar und zum Zeitpunkt der Ersterfassung als solche bekannt waren. Die Fortführung erfolgt laufend, sodass neue Straßen spätestens mit ihrer grundbücherlichen Durchführung im System nachgeführt werden.

5.1.4. Richtlinien für die Erfassung und Fortführung

- Rad- und Gehwege sind nur dann als eigene Straßenachsen abzubilden, wenn sie nicht Bestandteil einer übergeordneten Straßenanlage sind.
- Kanten unter einer Länge von einem Meter werden nicht dargestellt.
- Über- und Unterführungen werden als kreuzende Polylinien ohne Knoten abgebildet.
- Kanten dürfen sich nicht selbst überlagern.
- Pseudoknoten, die durch einen Attributwechsel gebildet werden, sind nur in folgenden Fällen zulässig:
 - Änderung der Straßengattung
 - Änderung der Straßenkategorie
 - Änderung der Straßenbezeichnung der Gemeinde
 - Änderung der Alternativbezeichnung
- Abbiegespuren von Kreuzungen und Kreisverkehrsfahrbahnen werden nur dann im Modell dargestellt, wenn die längste Kante der Anlage mehr als 50 Meter misst. In allen anderen Fällen bildet der Schnittpunkt der zulaufenden Straßenachsen den Kreuzungsmittelpunkt wie in der RVS 5.021 festgelegt.

5.1.5. Berücksichtigung der zeitlichen Komponente

Um Veränderungen im Straßennetz zu dokumentieren und die Konsistenz von Datensätzen, die den Straßengraphen als Bezugssystem verwenden, gewährleisten zu können, sind folgende Grundsätze zu beachten:

- Bei jeder neuen Kante sind die Benutzerkennung des Bearbeiters und das Entstehungsdatum in der Attributtabelle zu vermerken.
- Ungültige Straßenkanten werden nicht gelöscht, sondern nur durch das Einfügen eines Datumswertes im Feld *'Invalid'* gekennzeichnet. Zur Unterdrückung der Anzeige von ungültigen Kanten in ArcView ist der folgende Ausdruck im Feld der Themeneigenschaften erforderlich:
([Invalid] <= 0.AsDate)
- Werden Daten aus dem Straßensystem geschrieben scheint nur der momentan zeitlich aktive Stand auf. Sollen Daten, die durch eine Änderung des Bezugssystems ungültig geworden sind, wieder dargestellt werden, kann die Anzeigeeinschränkung wieder aufgehoben werden.

Bei den nachfolgend beschriebenen Änderungen am Straßengraphen (Werkzeuge der SIS-Extension) wird der jeweilige Stand vor der Manipulation durch das Inaktivsetzen der betroffenen Kanten gespeichert und die neu erzeugten Elemente mit dem aktuellen Datum und der Benutzerkennung versehen:



Teilung einer Kante infolge einer neuen Einbindung oder infolge eines Attributwechsels



Zusammenfügen einer Kante durch Wegfall eines Attributwechsels



Löschen einer Kante oder Veränderung der Kantenlage zwischen zwei Knoten

5.1.6. Logische Konsistenz des Modells

Zur Überprüfung der Netzstruktur wurden im Rahmen des Projektes SIS mehrere Avenue-Scripts entwickelt. Nachstehend dargestellte Fehler im Datenbestand sind mit Hilfe dieser Programme identifizierbar.

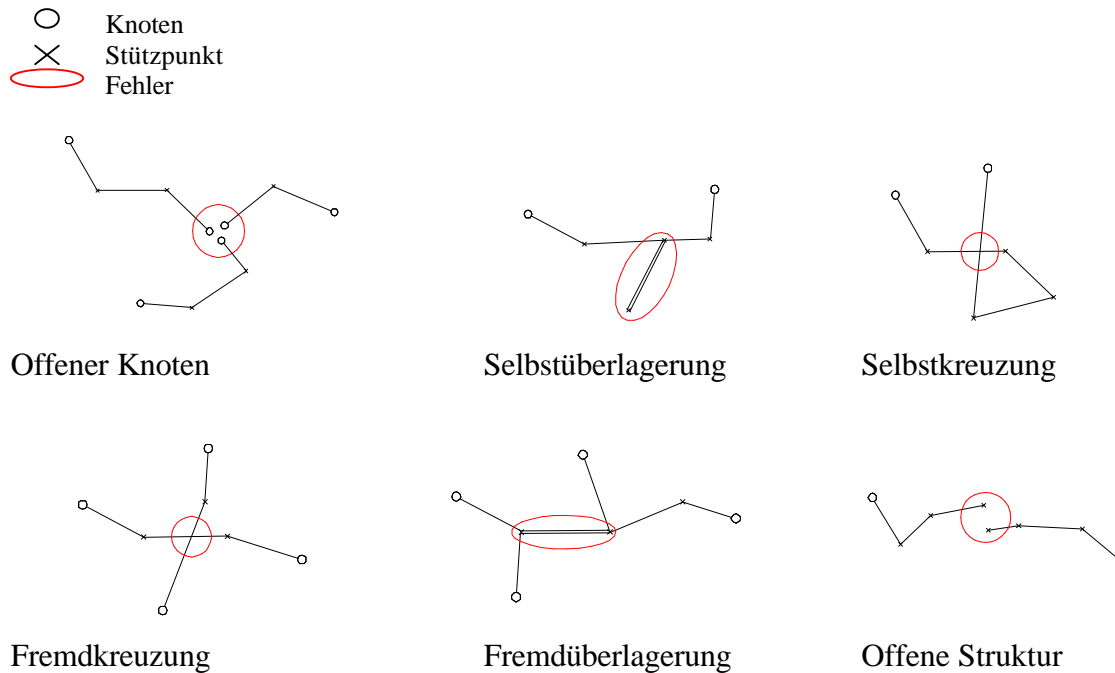


Abbildung 10: Fehlerarten

5.2. Datenmodell des Straßengraphen

Bei der Datenmodellierung wurde versucht sowohl die sachlichen Anforderungen nach dem O.ö. Straßengesetz 1991¹¹ zu erfüllen als auch die problemlose Verwendung der zur Verfügung stehenden Standardsoftwareprodukte zu gewährleisten.

Da Linien als Primitive nur immer einen Wert je definiertem Attribut enthalten können, ist bei jedem Attributwechsel ein Knoten (*Pseudoknoten*) einzufügen. Soll dieser Effekt generell verhindert werden, ist die Aufteilung in ein geometrisches Basissystem (nur Linien) und ein darauf aufbauendes getrenntes Ordnungssystem notwendig.

Beim SIS-Basissystem wurde die Zahl der Attribute so gewählt, dass der Straßengraph sowohl als begrenzter Informationsträger für das Straßenverzeichnis, als auch als Grundlage für Arbeiten mit dem Network Analyst dienen kann. Gleichzeitig sollen Linieninformationen, bei denen ein häufiger Attributwechsel zu erwarten ist, oder

¹¹ Landesgesetz vom 24. Mai 1991 über die öffentlichen Straßen mit Ausnahme der Bundesstraßen (LGBl. 84/1991)

solchen, die ein Ordnungssystem bedingen (Stationierungsrouten), einer dynamischen Segmentierung vorbehalten bleiben.

Die weitere Nutzung des Straßengraphen als Bezugssystem für Verwaltungsnetze bedingt die Vorhaltung von Topologieinformationen. Diese liegen in Form einer Knotentabelle (Punktthema) und der Beziehungen 'Startknoten' und 'Endknoten' in der Kantenattributtabelle vor. Die Topologieinformationen können mit den Werkzeugen der SIS-Extension nach Veränderung des Straßengraphen neu berechnet werden und stehen dem Anwender danach in transparenter Form zur Verfügung.

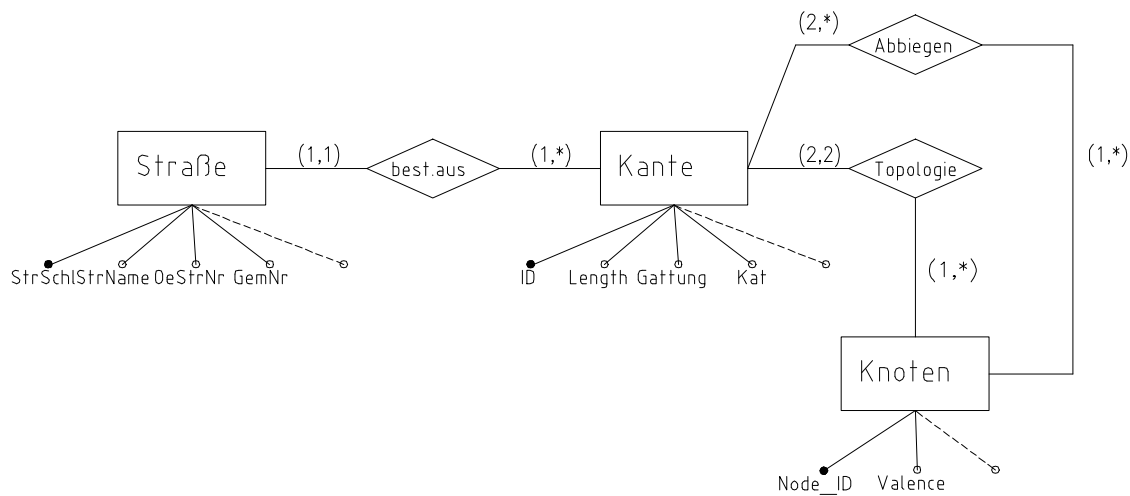


Abbildung 11: ER-Diagramm Straßengraph

5.2.1. Kanten

Die Abbildung des Straßengraphen erfolgt in Form einer ArcView-Shape Datei vom Typ 'Polyline'.

Attributtabelle:

Feldname	Datentyp	Breite	Präzision	Beschreibung
Shape	Shape/Polyline	9	0	Kante
ID	Number	16	0	Straßenschlüssel
Strnum	Number	11	0	Straßennummer Steyr lt. Straßenverzeichnis
Length	Number	16	2	Kantenlänge
Gattung	Number	2	0	Straßengattung
Kat	Number	2	0	Kategorie
FromNode	Number	12	0	Startknoten
ToNode	Number	12	0	Endknoten
Invalid	Datum	8	0	Kante inaktiv seit...
Speed	Number	5	0	Zulässige

				Höchstgeschwindigkeit
AltBez	String	10	0	Alternativbezeichnung z.B. B122
Oneway	String	2	0	Zulässige Fahrtrichtung
Vero	String	30	0	Straßenrechtliche Bewilligung
User	String	10	0	Benutzerkennung
Date	Datum	8	0	Erstellungsdatum

Mögliche Werte im Attribut 'Gattung':

- 1 Bundesstraße
- 2 Landesstraße
- 3 Gemeindestraße
- 4 Ortschaftsweg
- 5 Privatstraße

Mögliche Werte im Attribut 'Kat':

- 1 Straße
- 2 Radfahrweg
- 3 Fußgängerweg
- 4 Wanderweg
- 5 Güterweg

Mögliche Werte im Attribut 'Oneway':

- | | |
|-------|--|
| FT | Befahrbarkeit nur in Richtung der Kante |
| TF | Befahrbarkeit nur gegen die Kantenrichtung |
| N | Kante gesperrt |
| BLANK | Keine Einschränkungen |

Mögliche Werte im Attribut 'Strnum':

Neben den Straßennummern aus dem Straßenverzeichnis stellt die Zahl 9999 eine Kante außerhalb des Untersuchungsgebietes dar.

5.2.2. Knoten

Attributtabelle:

Feldname	Datentyp	Breite	Präzision	Beschreibung
Shape	Shape/Point	6	0	Knoten
Node_ID	Number	12	0	Knotenschlüssel
Valence	Number	12	0	Knotengrad
User	String	10	0	Benutzerkennung
Date	Datum	8	0	Erstellungsdatum

5.2.3. Straßenverzeichnis

Feldname	Datentyp	Breite	Präzision	Beschreibung
StrSchl	Number	4	0	Straßenschlüssel
Strname	String	30	0	Straßenbezeichnung
Oestrnr	Number	6	0	ÖSTAT-Straßenschlüssel
GemNr	Number	5	0	Gemeindennummer
GemBez	String	5	0	Gemeindebezeichnung
User	String	10	0	Benutzerkennung
Date	Datum	8	0	Erstellungsdatum

5.2.4. Abbiegeregeln (*turntables*)

Die ArcView-Extension 'Network Analyst' baut bei der Verwendung von Shape-Dateien nach Veränderungen am Netzwerk die Topologie für das zugrundeliegende Linienthema laufend neu auf. Als Schlüssel für die Knoten und Kanten wird dabei die jeweilige Datensatznummer verwendet. Problematisch ist dieser Umstand deshalb, weil sich Knoten- und Kantennummern mit jedem Eingriff in das Netzwerk verändern können. Für die Abbildung der Zeitdauer für den Wechsel von einer Kante zur nächsten (*Abbiegen*) sind es aber gerade diese Datensatznummern mit denen bei der Definition von Abbiegeregeln operiert werden muss. Auf diese Weise verlieren bei einem Eingriff in die Netzstruktur alle vorher erfassten Abbiegeregeln ihre Gültigkeit. Nun ist die Erhebung und Erfassung solcher Regeln in einem Stadtstraßennetz mittlerer Größe mit einigem Aufwand verbunden. Um eine dauerhafte Definition zu ermöglichen und die Regeln auch für die Netzwerkanalyse mit der ArcView-Extension einsetzen zu können, wurde der folgende Weg gewählt:

Wird nicht die Datensatznummer des Knotens, sondern seine Lage als Identifikationsmerkmal herangezogen, können die Abbiegerelationen in einem eigenen Punkthema in ArcView vorgehalten werden.

Datenstruktur des Abbiegethemas:

Feldname	Datentyp	Breite	Präzision	Beschreibung
Shape	Shape/Point	6	0	Knoten
ID	Number	12	0	Punkt ID (unabhängig vom jeweils aktuellen Knotenthema)
F_EDGE	Number	12	0	Von Kante
T_EDGE	Number	12	0	Nach Kante
SECONDS	Number	12	0	Abbiegedauer 'Von Kante' – 'Nach Kante' in Sekunden (Abbiegen unzulässig = -1)
User	String	10	0	Benutzerkennung
Date	Datum	8	0	Erstellungsdatum

Die Attributtabelle enthält dabei die Kantennummern des Straßengraphen.

Für die weitere Verwendung der Informationen muss zwischen der Topologiestruktur, die im Straßeninformationssystem (SIS) zur Verfügung steht, und jener die von der Network-Analyst-Extension erzeugt wird, unterschieden werden:

Beim Aufbau von Topologieinformationen im SIS wird automatisch auch ein Punktthema generiert, in dem die Nummer eines jeden Knotens und die Zahl der dort eingebundenen Kanten abgebildet wird. Die Zuordnung der Knotennummer kann in diesem Fall relativ einfach, durch einen räumlichen 'Join' beider Punktthemen erfolgen.

Die Lösung zur Nutzung der Informationen mit der Network-Analyst-Extension (NA) gestaltet sich dabei um einiges komplizierter.

Die Erweiterung verwendet, wie bereits erwähnt, in der Abbiegetabelle keine gleichbleibenden Zählerwerte als Knoten- und Kantenschlüssel, sondern nur die aktuellen Datensatznummern. Das ist auch der Grund dafür, dass hier eine neue Tabelle mit Abbiegebeziehungen zu generieren ist, die auf der Position der Datensätze und nicht auf Tabellenwerten basiert.

Das Avenuescript *SIS.MkTurnTable* bildet dazu aus dem SIS-Abbiegethema eine mit den Daten aus dem *network index directory* (Anm. Topologieinformationen des Network Analyst) korrespondierende Tabelle. Es folgt eine kurze Ablaufbeschreibung des Algorithmus:

- Auswählen und Überprüfen des Abbiegethemas
- Auswählen und Überprüfen des Linienthemas
- Sicherstellen der NA-Dateistruktur für das Linienthema
- Erzeugen einer neuen Tabelle mit den erforderlichen Feldern
- Einlesen SIS-Abbiegedatensatz
 - Selektieren der Kanten F_EDGE und T_EDGE im Straßengraphen aufgrund der Tabellenwerte
 - Bestimmen der Recordpositionen der Kanten und der Koordinaten der jeweiligen Anfangs- und Endpunkte.
 - Holen der Einträge der Knotennummern beider Kanten aus der NA-Netzwerktable NODE.DBF über die Recordnummern der Kanten.
 - Speichern der Nummern und der dazugehörigen Koordinaten in einer Avenue Dictionarystruktur der folgenden Form: {Point, JUNCTION}
 - Suche den Punkt aus dem Abbiegethema in der erzeugten Liste mit Hilfe seiner Position
 - Wird der Punkt nicht gefunden, dann ist eine Fehlermeldung auszugeben und mit dem nächsten Datensatz fortzusetzen.
 - Kopiere den Wert SECONDS aus der Abbiegetabelle und schreibe die Recordnummern der Kanten und den gefundenen Wert von JUNCTION in einen neuen Datensatz der NA-Abbiegetabelle.
- Wenn das Tabellenende erreicht wurde abbrechen, sonst zum nächsten Datensatz.

Beim Testen des Programms stellte sich heraus, dass die Datensätze nach Knotennummern geordnet, in der Tabelle abgelegt sein müssen. Abhilfe konnte mit dem ESRI-Userscript *Table.SortPhysical* von W.HUBER¹² geschaffen werden, mit dem das physische Sortieren von Tabelleninhalten ermöglicht wird.

¹² W. Huber, Quantitative Decisions, Merion, PA USA 6/11/98, <http://gis.esri.com/arcscrips>

Die Einbindung von Abbiegebeziehungen bei Verwendung der Network-Analyst-Extension wird hier nicht gesondert erläutert. In diesem Zusammenhang darf auf die Online-Hilfe der Anwendung verwiesen werden (*sh. network turntables*).

5.3. Straßenverzeichnis und Plandarstellung

Die Anforderungen betreffend den Inhalt von Straßenverzeichnissen wurden bereits beschrieben. Für die laufende Wartung und Abfrage von Informationen über einzelne Abschnitte des Stadtstraßennetzes steht ein Standardformular innerhalb der SIS-Oberfläche zur Verfügung.

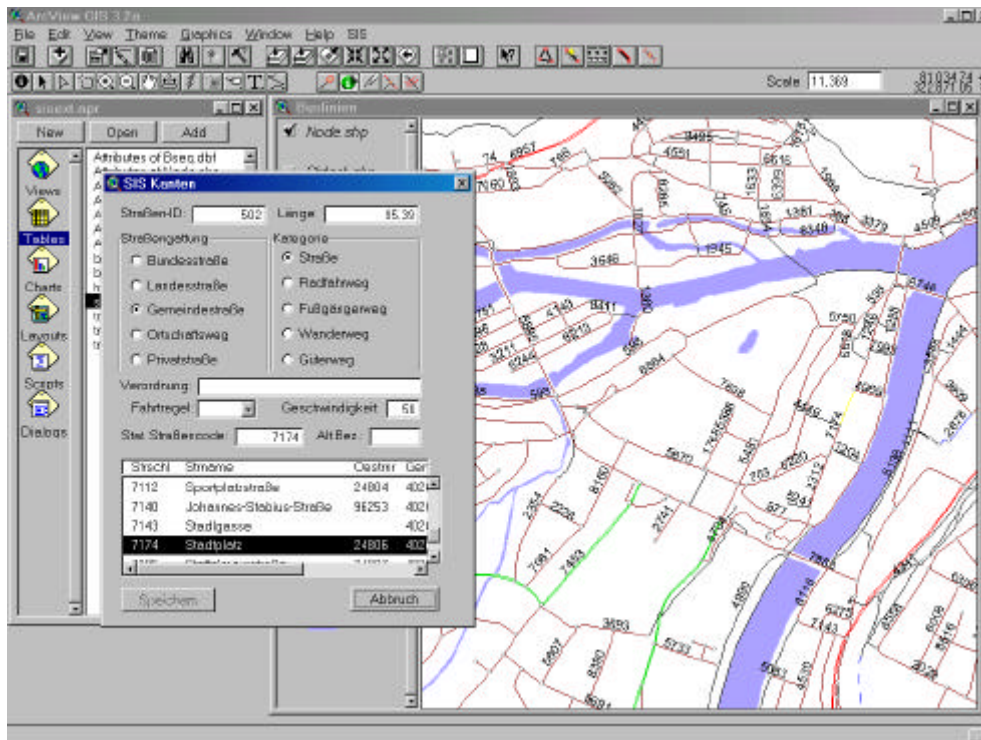


Abbildung 12: Oberfläche der SIS-Erweiterung mit Kantenattributdialog

Zusammengefasst lauten die gesetzlichen Anforderungen an die Plandarstellung der Straßenkarte wie folgt:

- Maßstab 1:20.000 (nicht zwingend) für das gesamte Gemeindegebiet mit Ausweisung der Gattung und einer zugeordneten Zahl.
- Gemeindestraßen braun
- Ortschaftswege gelb
- Güterwege orange
- Radwege strichpunktiert schwarz
- Fußgänger- und Wanderwege schwarz

Ergänzende Angaben:

- Der Maßstab für die Straßenkarte wird mit 1:10.000 gewählt, um die vorgeschriebenen Informationen in geeigneter Weise darstellen zu können.
- Bundesstraßen werden rot, Landesstraßen grün dargestellt.
- Private Straßenanlagen werden blau abgebildet.

Die Straßenkarte der Stadtgemeinde Steyr im Maßstab 1:10.000 ist Bestandteil der Projektarbeit und liegt dieser als Anhang A bei.

Straßenstatistik der Stadt Steyr mit Stand Dezember 2001:

Straßengattung (O.ö. Straßengesetz 1991)	Länge in Kilometern	Abschnitte
Bundesstraßen	17,49	99
Landesstraßen	8,15	49
Gemeindestraßen	179,42	1567
Radfahrwege	1,35	5
Fußgängerwege	31,61	266
Privatstraßen	3,08	43
Privatwege	0,81	7
Summe	241,91	2036

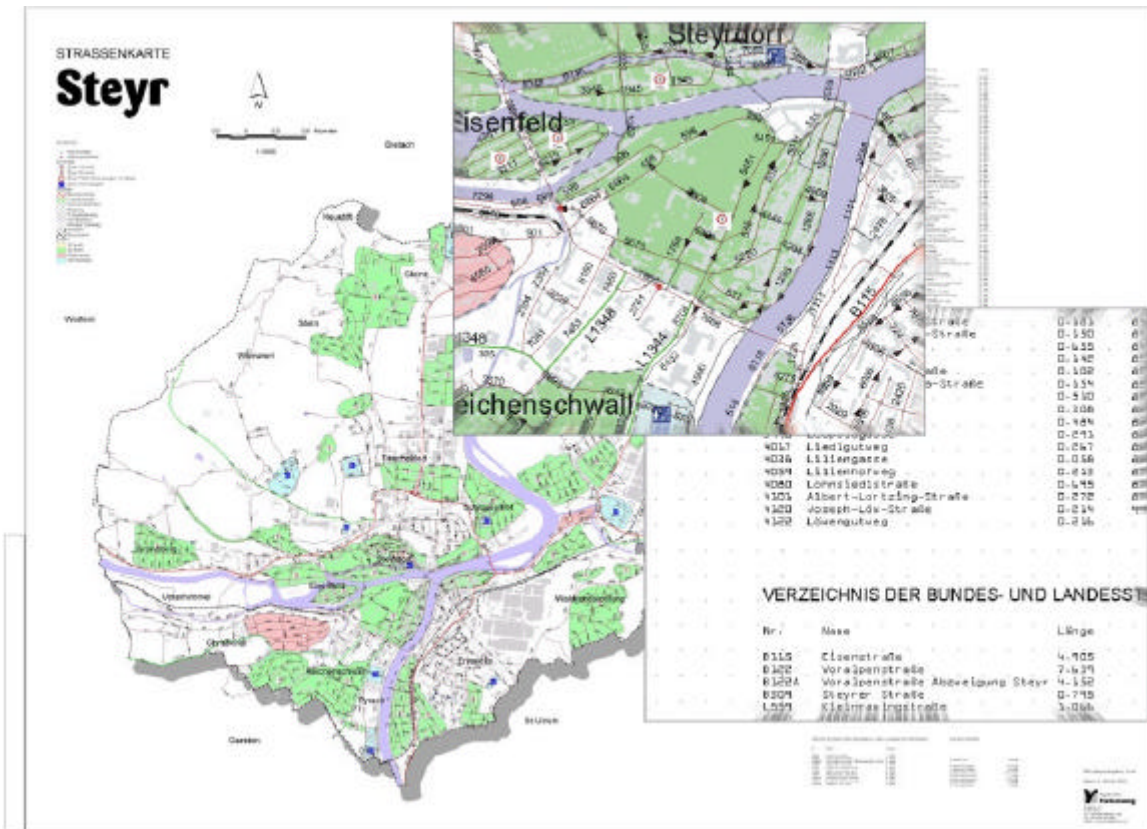


Abbildung 13: Verkleinerte Darstellung der Straßenkarte

6. Datenmodell für die dynamische Segmentierung

Die nach dem Modell der "Dynamischen Segmentierung" auf dem Straßengraphen abgebildeten Routen, stellen entweder (gerichtete) lineare Ereignisse dar oder bilden selbst Teile von neuen Verwaltungsnetzen. Die Routen setzen sich dabei aus geordneten, stationierten Abschnitten auf dem Straßengraphen zusammen.

Für die vorliegende Projektarbeit wurde das Busliniennetz von Steyr als konkretes Beispiel für ein solches Verwaltungsnetz gewählt. Auch der Aufbau eines Bezugssystems für Straßen, wie in der RVS 5.011 definiert, kann mit Hilfe dieses Modells bewerkstelligt werden. Die Umsetzung dieser Richtlinie muss aber aufgrund des zeitlichen Aufwandes einem zukünftigen Projekt vorbehalten bleiben.

Im Zusammenhang mit der RVS wird auf den Umstand hingewiesen, dass im nachfolgenden Text punktuelle Informationen auf dem Straßengraphen, in einer von der RVS abweichenden Definition, als *Bezugspunkte* bezeichnet werden.

Für die Abbildung von Punkten und Routen auf dem Verwaltungsnetz wird das Datenmodell des Straßengraphen wie folgt erweitert:

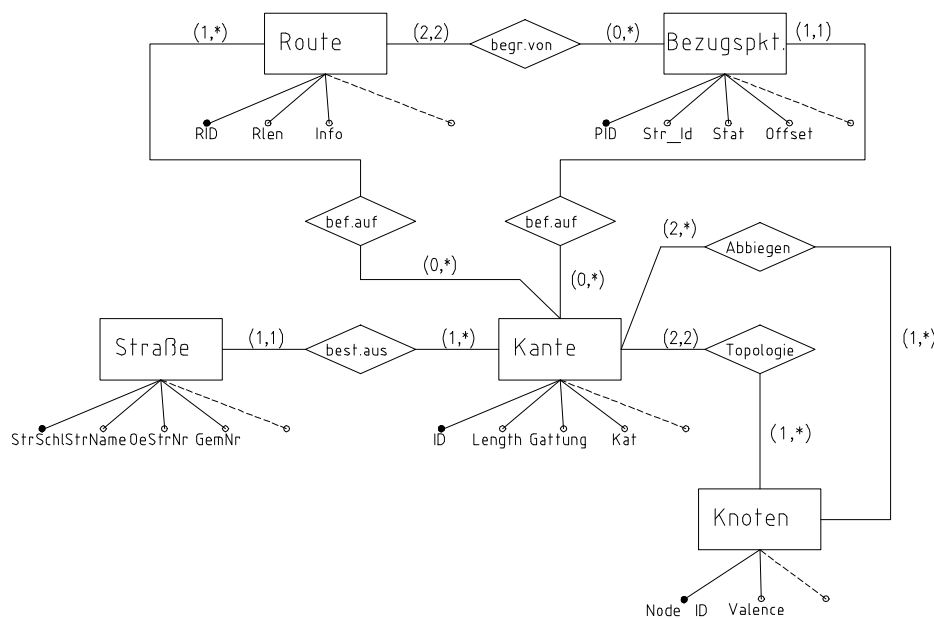


Abbildung 14: ER-Diagramm Routen und Bezugspunkte

Die Speicherung der Daten erfolgt in dBASE-Dateien. Für die Anzeige der Daten in ArcView wird ein für diese Zwecke eigens aufbereitetes Linienthema vom Shapetyp *PolylineM* verwendet.

Bezugselemente zur räumlichen Orientierung von Linien- und Punkthemen auf dem Straßengraphen sind der eindeutige Kantenschlüssel und die Stationierung auf der Kante. Die Verortung von Punkten auf dem Straßengraphen kann mit Hilfe der SIS-Werkzeuge manuell oder automatisch erfolgen. Für die Routenbildung ist die Auswahl der Straßenkanten durch den Benutzer erforderlich. Werden lineare Informationen als Route abgebildet, ist die Vorhaltung der Richtungskomponente unerlässlich. Erfolgt die Erfassung einer Route immer nur zwischen zwei Bezugspunkten, kann nicht nur eine einfache und für den Benutzer transparente Orientierung der Route erfolgen (*von BPI*

nach BP2). Es ist so auch möglich bereits bei der Erfassung eine eigenständige Topologie mit aufzubauen.

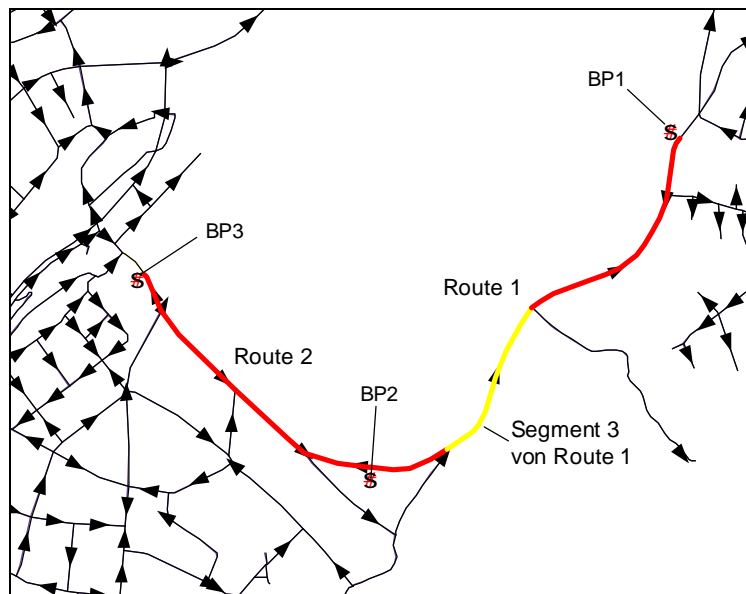


Abbildung15: Beschreibung einer Route

Ein Routenthema wird im SIS-Steyr im Normalfall durch drei Tabellen beschrieben. Einer Bezugspunkt-, einer Routen- und einer Segmenttabelle. Die Routentabelle enthält neben einer Routenkennung noch Angaben über die Gesamtlänge und den Verlauf der Route zwischen den Bezugspunkten (Routentopologie). Erfolgt die Routenbildung ohne Bezugspunkte, sind die Start- und Endkante zur Gänze Bestandteil der Route. In einem solchen Fall enthalten die Felder der Routentopologie (*BPFrom* und *BPTo*) den Wert -1. Die Segmenttabelle wird zur Abbildung der Route auf dem Bezugssystem verwendet. In ihr sind alle Routenteile mit ihrem Raumbezug enthalten. Die Kantenabschnitte werden in der Segmenttabelle geordnet (Attribut *Order*), orientiert (Attribut *Direction*) und mit einer eigenen Routenstationierung versehen abgelegt.

Die Prüfung der Route auf Lücken, Kreuzungen oder unzulässige Abzweigungen erfolgt zum Zeitpunkt der Bildung durch eine Prüfung der Knotenliste mit Hilfe der Regel:

$$\text{Anzahl der Innenknoten} = \text{Anzahl der Kanten} - 1$$

Durch die Anwendung der Vorschrift können folgende Fehler bei der Routenbildung vermieden werden:

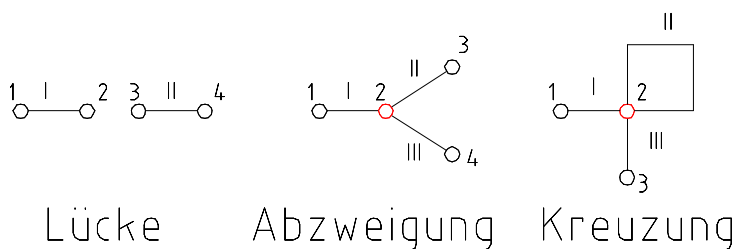


Abbildung 16: Mögliche Routenfehler (die Innenknoten wurden rot dargestellt)

Die Innenknoten werden beim Test durch ihr mehrfaches (im Normalfall doppeltes) Auftreten in der Kantentopologie identifiziert.

Für die Reihung der Routensegmente wird aus den Topologieinformationen der selektierten Kanten zuerst ein gerichteter Graph in Form einer Speicherliste gebildet. Anschließend wird die Route ausgehend von der Startkante bis zur Endkante durchlaufen und die Abschnitte geordnet in der Segmenttabelle gespeichert.

Den einzigen Fall einer zulässigen "Kreuzung" von Bezugskanten bildet die Einbindung der Start oder Endkante in den Routenverlauf. In solchen Fällen wird der Benutzer zur Eingabe darüber aufgefordert, ob die Route an diesen Abschnitten in oder gegen die Kantenrichtung verlaufen soll. Im Routenbildungsprogramm werden zur Behandlung solcher Ausnahmen die jeweiligen Knotennummern des gerichteten Graphen mit ungültigen Werten ausmaskiert.

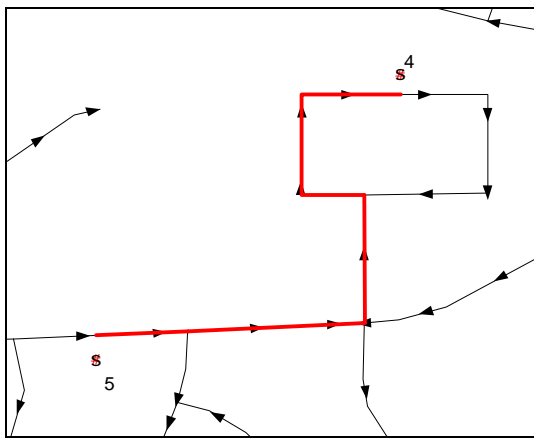


Abbildung 17: Sonderfall einer Route

6.1. Bezugspunkte

Feldname	Datentyp	Breite	Präzision	Beschreibung
PID	Number	16	0	Punktschlüssel
STAT	Number	16	2	Stationierungswert in Metern
OFFSET	Number	16	2	Abstand von der Bezugskante in Metern
STR_ID	Number	16	0	Kantenschlüssel der Bezugskante
INFO	String	30	0	Benutzeridentifikation
X	Number	16	2	Rechtswert
Y	Number	16	2	Hochwert
User	String	10	0	Benutzerkennung
Date	Datum	8	0	Erstellungsdatum

6.2. Routen

Routentabelle:

Feldname	Datentyp	Breite	Präzision	Beschreibung
RID	Number	16	0	Routenschlüssel
BPFROM	Number	16	0	Von Bezugspunkt (-1 bei Routenbildung ohne BP)
BPTo	Number	16	0	Nach Bezugspunkt (-1 bei Routenbildung ohne BP)
RLEN	Number	16	2	Länge der Route
INFO	String	30	0	Benutzeridentifikation
User	String	10	0	Benutzerkennung
Date	Datum	8	0	Erstellungsdatum

Segmenttabelle:

Feldname	Datentyp	Breite	Präzision	Beschreibung
SEGID	Number	16	0	Segmentschlüssel
RID	Number	16	0	Routen ID
STRID	Number	16	0	Straßen ID
STATFROM	Number	16	2	Stationsanfang Bezugskante in Metern
STATTO	Number	16	2	Stationsende Bezugskante in Metern
ORDER	Number	16	0	Reihungszahl des Segments beginnend mit 0
DIRECTION	Number	2	0	Orientierung des Segments, bei Verlauf in Richtung der Bezugskante 1 sonst -1
MSFROM	Number	16	2	Stationsanfang des Segments in Metern
MSTO	Number	16	2	Stationsende des Segments im Metern
User	String	10	0	Benutzerkennung
Date	Datum	8	0	Erstellungsdatum

7. Befehlsumfang der SIS-Erweiterung

Die SIS-Umgebung enthält neben Werkzeugen zur Bearbeitung und Prüfung des Straßengraphen auch Befehle für die GIS-unterstützte Erfassung von dynamisch segmentierter Information.

Die Programmentwicklung erfolgte mit der in ArcView integrierten Scriptsprache Avenue. Die beschriebenen Hilfsmittel wurden in der ArcView-Extension *sis.avx* zusammengefasst. Neben der Möglichkeit zur problemlosen nachträglichen Einbindung in bestehende Projekte, kann damit auch die Installation der Anwendung erleichtert werden. Die in der nachfolgenden Beschreibung mit Fußnoten gekennzeichneten Befehle, arbeiten mit User-Scripts aus dem Downloadbereich der ESRI-Homepage. Alle anderen Programmsequenzen wurden im Rahmen der Projektarbeit entwickelt. Zur besseren Orientierung wurde neben dem Titel die Themenzugehörigkeit der Befehle in Klammer angegeben.

Die verwendeten Avenuescripts sind im Anhang B dokumentiert.

7.1. SIS-Funktionen in der Menüleiste

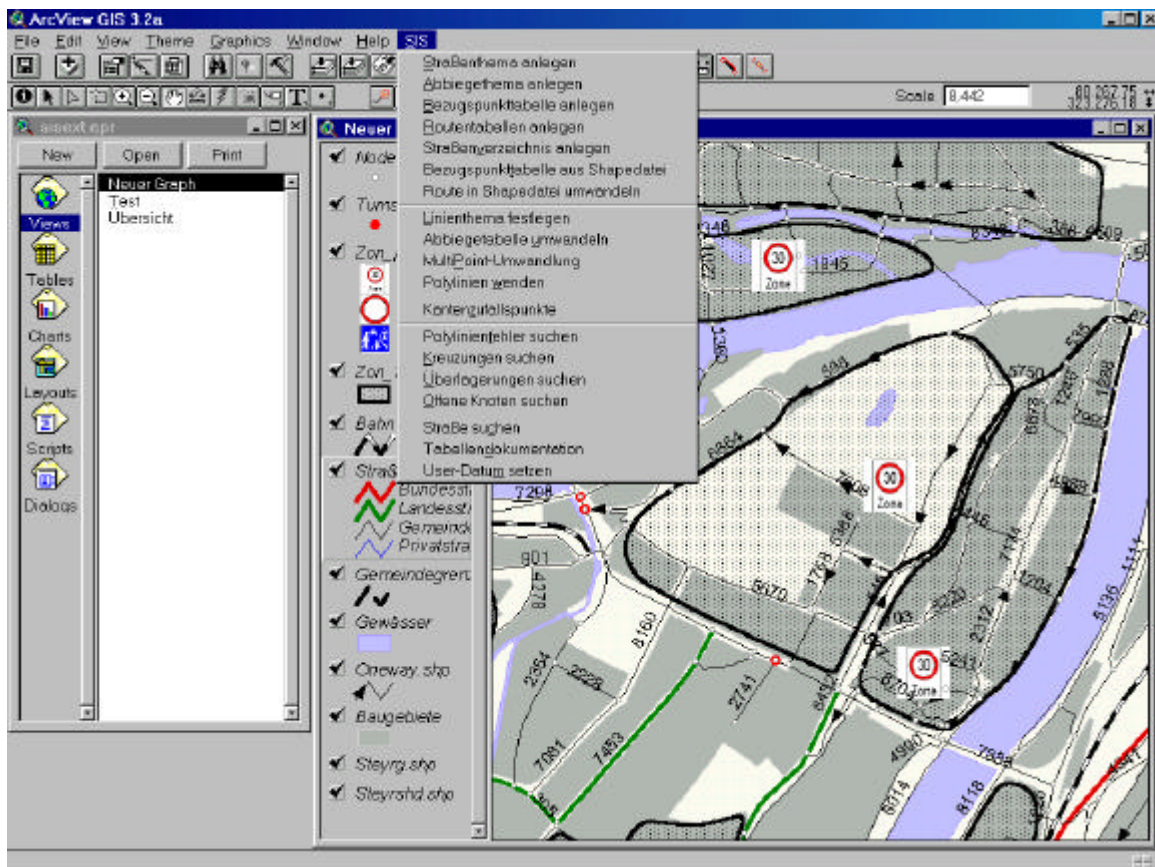


Abbildung 18: Menüstruktur

Straßenthema anlegen (Straßengraph)

Erstellen eines neuen Linienthemas mit den in Punkt 5.3.2. definierten Attributfeldern. Der so definierter Straßengraph dient in weiterer Folge als Bezugssystem und Betriebsnetz im SIS.

Abbiegethema anlegen (Straßengraph)

Hier wird ein neues Punktthema erstellt, mit dem die Abbiegeregeln zu einem Straßengraphen definiert werden können.

Bezugspunktabelle anlegen (dyn. Segmentierung)

Der Befehl erzeugt eine Tabelle mit allen, für die manuelle Erfassung von Bezugspunkten auf dem Straßengraphen erforderlichen Feldern.

Routentabellen anlegen (dyn. Segmentierung)

Mit Hilfe dieser Funktion werden die beiden für ein Routenthema erforderlichen Tabellen erzeugt.

Straßenverzeichnis anlegen (Straßengraph)

Erstellt eine Tabelle zur Erfassung von Straßennamen (*strverz.dbf*).

Bezugspunktabelle aus Shapedatei (dyn. Segmentierung)

Punktthemen können mit diesem Befehl nach der Methode der dynamischen Segmentierung automatisch auf ein bestehendes Linienthema projiziert werden. Voraussetzung ist die eindeutige Kantenbezeichnung im verwendeten Linienthema. Ergebnis ist eine neue Tabelle mit allen Attributwerten des Ausgangsthemas und den Bezugselementen (Kanten-ID, Station) auf dem Straßengraphen. Das zugrundeliegende Avenuescript *SIS.PtToLineTbl* stellt zunächst einen räumlichen Join zwischen Punkten und Linien her. Anschließend werden die Kanten für jeden Punkt auf Basis ihrer Bezeichnung im Linienthema gesucht. Die Stationierungswerte werden berechnet und anschließend in der neuen Tabelle gespeichert.

Route in Shapedatei umwandeln (dyn. Segmentierung)

Mit diesem Befehl können dynamisch segmentierte Linieninformationen in eine Shapedatei vom Typ *Polyline* umgewandelt werden. Die Segmente werden dabei richtig orientiert und auf Basis des Wertes im Feld RID (Routenkennung) zusammengefasst. Die zur Segmenttabelle gehörige Routentabelle ist vom Benutzer anzugeben.

Linienthema festlegen (dyn. Segmentierung)

Der Menüpunkt ermöglicht die Festlegung jener Tabellen, die in weiterer Folge bei der Erfassung von Routen verwendet werden sollen. Auszuwählen sind das Bezugspunkt- und das Bezugslinienthema (im aktuellen View), sowie jene Routen- und Segmenttabelle die für die Speicherung der Informationen vorgesehen sind.

Abbiegetabelle umwandeln (Straßengraph)

Die Funktion erzeugt aus einem SIS-Abbiegethema eine Tabelle, die als *turntable* mit dem Network Analyst verwendet werden kann.

Anm.: Die in diesem Punkt beschriebene Funktionalität benötigt ein für das Linienthema (Shapedatei) gültiges Netzwerk-Index Verzeichnis (network index directory, *.nws). Diese Struktur wird zwar vom Network Analyst erzeugt, die ESRI-Extension selbst, ist für keine der beschriebenen SIS-Funktionalitäten erforderlich.

Multipoint-Umwandlung¹³ (dyn. Segmentierung)

Werden Punkte mit Hilfe der dynamischen Segmentierung in ArcView auf ein Linienthema aufgetragen und anschließend in ein Punkthema umgewandelt, entsteht ein Thema vom Shapetyp *MultipointM*. Methoden der Punkt-Klasse, wie beispielsweise *Shape.GetX* (zur Bestimmung des Rechtswertes), sind hier nicht anwendbar. Ebenso bereitet dieser Shape-Typ im Umfeld von Arc-IMS V 1.0 Probleme. Mit Hilfe des Menüpunktes kann aus ausgewählten GIS-Datensätzen eine herkömmliches Punkthema erzeugt werden.

Polylinien wenden¹⁴ (Straßengraph)

Da die vorgegebene Fahrtrichtung im Straßengraphen, unabhängig von seiner eigentlichen Orientierung durch einen Attributwert festgelegt wird, besteht grundsätzlich kein Bedarf, Änderungen an der Linienstruktur des Straßengraphen vorzunehmen. Die Funktion wurde aus der Notwendigkeit heraus integriert, die korrekte Darstellung von Einbahnen in Planwerken zu ermöglichen.

Kantenzufallspunkte (Straßengraph)

Der Befehl ist bereits aus dem Kapitel über die Genauigkeit des Straßengraphen bekannt. Nach Eingabe einer gewünschten Zahl von Zufallspunkten werden aus dem aktuellen Linienthema (Shapetyp *Polyline*) nach dem Zufallsprinzip (Avenuefunktion: *MakeRandom*) Linien ausgewählt, in deren Mitte dann ein Grafikpunkt erzeugt wird. Die Umwandlung in eine Shapedatei kann durch das Kopieren und Einfügen der Grafikelemente in ein Punkthema erfolgen.

Polylinienfehler suchen (Straßengraph)

Die Programmsequenz prüft jedes einzelne Liniensegment auf Selbstüberlagerung, Selbstkreuzung und offene Linienstrukturen (sh. auch Punkt 5.2.6., Logische Konsistenz des Modells)

Fehlerfiguren:

<u>Typ</u>	<u>Element</u>	<u>Farbe</u>
Selbstüberlagerung	Linie	blau
Selbstkreuzung	Punkt	rot
Offene Linienstruktur	Linie	grün

Kreuzungen suchen (Straßengraph)

Die Funktion ermöglicht es, das aktuelle Linienthema auf unzulässige Kreuzungen zu untersuchen. Die angezeigte Fehlerfigur an Kreuzungen ist ein roter Punkt.

Überlagerungen suchen (Straßengraph)

Wird eine Kante des aktuellen Themas von einer anderen Kante im selben Thema überlagert, so wird dies bei Ablauf der vorliegenden Prüfsequenz durch eine blaue Linie angezeigt.

¹³ Heiko Apel, Technical University of Braunschweig, Germany, <http://gis.esri.com/arcscripsts>

¹⁴ Michael Herter 07.12.1999, <http://gis.esri.com/arcscripsts>

Offene Knoten suchen (Straßengraph)

Datengrundlage für die Nutzung der Funktion ist ein Punktthema, mit den aus der Topologieberechnung stammenden Netzwerkknoten (sh. auch die Funktion Netzwerk). Beim Aufruf der Routine ist vom Anwender ein Schwellenwert anzugeben, der den Mindestabstand zwischen zwei Knoten festlegt. Das Prüfprogramm ermittelt dann nicht nur Liniensegmente mit einer Länge unter dem Schwellenwert, sondern bestimmt auch fehlerhaft eingebundene Kanten.

Fehlerfiguren sind jeweils zwei grüne Punkte an den fehlerhaften Knoten.

Straße suchen (Straßengraph)

Die Funktion dient dem Auffinden von Straßenzügen im Straßengraphen. Da die Straßennamen nicht im Klartext im Straßenthema gespeichert sind, sondern nur über den Straßencode, besteht hier (bei Vorliegen der Tabelle *strverz.dbf* mit einem gültigen Straßenverzeichnis) die Möglichkeit Straßennamen, ohne einen vorhergehenden 'Join' beider Tabellen zu suchen. Nach Eingabe eines Textteiles aus dem Straßennamen wird das aktuelle View nach dem ersten Linienthema mit dem numerischen Feld 'Strnum' durchsucht. Werden anschließend im Straßenverzeichnis mehrere Straßen mit gleichen Teilzeichenfolgen entdeckt, kann der Anwender den gewünschten Namen in einer Dialogbox auswählen. War die Suche erfolgreich wird der View-Ausschnitt so gewählt, dass die gesuchten Elemente vollständig sichtbar sind.

Tabellendokumentation¹⁵

Der Befehl zeigt den Aufbau einer im aktuellen Projekt eingebundenen Tabelle an und ermöglicht das Speichern der Strukturinformationen in einer Textdatei.

User-Datum setzen

Beim Laden der SIS-Extension wird der Benutzer zur Eingabe seiner Benutzerkennung aufgefordert. Mit dem Menüpunkt '*User-Datum setzen*' können ausgewählte Datensätze von editierbaren Themen mit dieser Kennung und dem aktuellen Datum versehen werden. Dabei werden die bestehenden Werte in den Feldern *User* und *Date* überschrieben. Sind die Felder in der Attributtabelle noch nicht vorhanden, werden sie vorher automatisch erzeugt. Wurde kein Element ausgewählt, werden alle Datensätze als Auswahl Satz verwendet.

7.2. SIS-Funktionen der Button-Leiste

Netzwerk¹⁶ (Straßengraph und Dyn. Segmentierung)

Die Funktion ermöglicht den Aufbau einer Netzwerktopologie aus dem aktuellen Linienthema. Falls noch nicht vorhanden, wird die Attributtabelle des Linienthemas um die Felder *FromNode* und *ToNode* erweitert. Das Generieren eines eigenen Punktshapes mit den korrespondierenden Knoten ist ebenfalls möglich.

Wird ein Linienthema als Basis für die dynamische Segmentierung verwendet, ist die vorherige Topologiebildung mit der beschriebenen Funktion zwingend erforderlich. Die Attribute *FromNode* und *ToNode* werden für die Prüfung der projizierten Linienstruktur benötigt.

¹⁵ RAZAVI A.H. und WARWICK V.(1997), ArcView Gis/Avenue Programmer's Reference, OnWord Press, Camino Entrada

¹⁶ SALGARO R., TRDIN A. et al. (2000), <http://gis.esri.com/arcscrippts>

Linienthema aktualisieren (Straßengraph)

Nach einer Änderung oder Nachführung des Straßengraphen können mit dieser Funktion die Werte für die Kantenschlüssel ergänzt werden (ArcView kennt keine Zählervariablen). Zusätzlich werden die Kantenlänge, der Wert für die Kantenüberwindungsdauer und der kombinierte Symbolschlüssel für die Signatur der Kanten neu berechnet. Gültige Kantenschlüssel sind Voraussetzung für die dynamische Segmentierung.

Linienthema für Dynamische Segmentierung erzeugen¹⁷

ArcView bietet seit der Version 3.1 die Möglichkeit auch Shape-Dateien als Basis für die dynamische Segmentierung zu verwenden. Mit dem vorliegenden Befehl besteht die Möglichkeit aus dem aktiven Linienthema ein solches Thema vom Shapetyp *PolyLineM* zu erzeugen. Sollen Routen oder Punkte auf dem Netzwerk erfasst werden, ist auch der Straßengraph, der als herkömmliches Polylinienthema vorliegt, mit dieser Funktion vorher umzuwandeln.

Route definieren (Dyn. Segmentierung)

Zur Bildung einer Route, sind ein geeignetes Linienthema (sh. Netzwerk), ein darauf aufbauendes Bezugspunktthema, sowie je eine Routen- und eine Segmenttabelle erforderlich. Das Bezugspunktthema muss dabei ein mit Hilfe der dynamischen Segmentierung auf ein Linienthema projiziertes Punktthema sein. Die Routentabellen können mit der Menüfunktion 'Routentabellen anlegen' erzeugt werden. Welche Themen und Tabellen von der Funktion verwendet werden ist von den Einstellungen abhängig, die zuvor im Menüpunkt 'Linienthema festlegen' vorgenommen wurden. Ohne vorherige Definition wird die Funktion mit einer Fehlermeldung abgebrochen. Um eine Route erfassen zu können, sind bereits vor dem Aufruf der Funktion, die Bezugspunkte und alle dazwischenliegenden Kanten auszuwählen. Da die Route auf einen unzulässigen Verlauf hin überprüft wird, muss das zugrunde liegende Linienthema über gültige Topologieinformationen verfügen (sh. auch die Menüpunkte *Netzwerk* und *Linienthema für Dynamische Segmentierung erzeugen*). Wird ein korrekter Linienzug erkannt, wird der Benutzer aufgefordert, die Orientierung der Route durch die Wahl des Startbezugspunktes oder der Startkante festzulegen, wenn keine Bezugspunkte ausgewählt wurden.

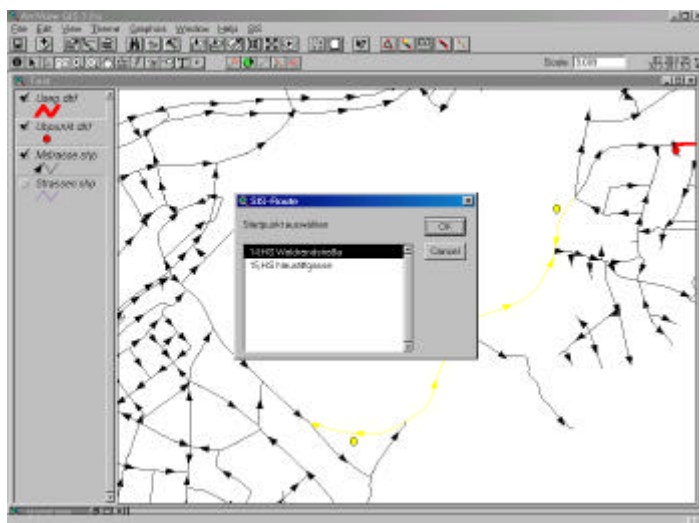


Abbildung19: Definition einer Route

¹⁷ ArcView 3.2 Online Help

In der Dialogbox zur Wahl des Startelements werden der Punktschlüssel und das Attribut *Info* aus der Bezugspunktabelle oder die Kanten-ID des Straßengraphen angezeigt. Die Route selbst darf keine Kreuzungen enthalten. Die einzige Ausnahme bilden mögliche Einbindungen von Start- oder Endkante in die Route. In solchen Fällen wird der Anwender nach der Orientierung der Route in Bezug auf die Kante befragt. Abschließend kann eine durch den Benutzer frei wählbare Bezeichnung für die Route eingegeben werden. Bestätigt der Anwender seine Eingabe wird ein neuer Datensatz angelegt. Beginnt oder endet eine Route an einem Bezugspunkt mit einem Stationswert von 0 (*Startknoten*) oder einem Stationswert, welcher der Kantenlänge entspricht (*Endknoten*), ist auch jene Kante, auf die der Bezugspunkt verweist, auszuwählen.

Kanten zusammenfügen (*Straßengraph*)

Ist ein Linienthema zur Bearbeitung geöffnet, besteht mit Hilfe dieses Befehls die Möglichkeit zwei Kanten zusammenzufügen. Wie in Punkt 5.1.5. beschrieben, werden die gewählten Kanten vor der weiteren Bearbeitung als ungültig gekennzeichnet. Danach werden Kopien der betroffenen Kanten zusammengefügt und diese dem Linienthema neu hinzugefügt. Um Fehlfunktionen zu vermeiden, ist es unbedingt erforderlich eine jede Bearbeitung abzuschließen und zu speichern. Die Unterdrückung der Anzeige ungültig gewordener Kanten erfolgt nämlich erst nach dem Beenden und Speichern der Änderungen.

7.3. SIS-Funktionen der Werkzeugleiste

Bezugspunkt setzen (*Dyn. Segmentierung*)

Basis für die Befehlsausführung bildet ein gültiges Linienthema vom Typ *PolylineM* im aktuellen View. Das Linienthema muss dabei zwingend ein numerisches Feld mit dem Schlüsselattribut ID enthalten.

Nach Anlegen einer Bezugspunktabelle mit der SIS-Menüfunktion '*Bezugpunkttabelle anlegen*' und deren anschließender Definition als Ereignisthema (*Event Theme*) im aktuellen View, können Punkte auf dem Straßengraphen verortet werden. Die räumliche Definition erfolgt, indem der Bezugspunkt mit der Maus im View festgelegt wird und die zugehörige Kante, ohne die linke Maustaste dabei loszulassen, mit dem Mauszeiger geschnitten wird. Die so ermittelten Stationierungswerte können nachträglich in der Maske '*SIS-Bezugspunkt*' noch verändert werden.

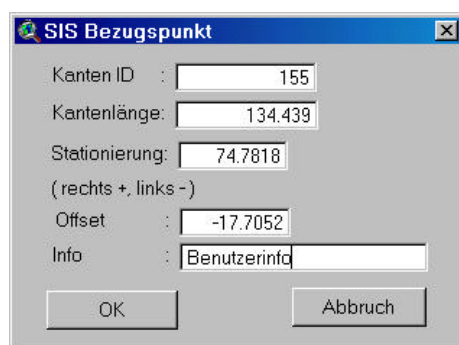


Abbildung 20: Dialogbox für die Erfassung von Bezugspunkten

Routenelement löschen (Dyn. Segmentierung)

Die Datensätze der ausgewählten Route werden dabei sowohl in der Segment-, als auch in der Routentabelle gelöscht (Anm.: Derzeit erfolgt bei Routen die Berücksichtigung eines Zeitfaktors nur dahingehend, dass das Erfassungsdatum festgehalten wird). Werden bei der Auswahl mehrere übereinander liegende Routen identifiziert, besteht mit Hilfe einer Dialogbox die Möglichkeit zur Auswahl der gewünschten Strecke.

Anm.: Das Routenthema muss vorher mit dem Menüpunkt '*Linienthema definieren*' richtig festgelegt worden sein.

Kreuzung einfügen (Straßengraph)

Die Funktion ermöglicht die Erweiterung des Straßengraphen durch Einfügen einer neuen Kante. Jene Kanten, die von der neuen Linie geschnitten werden, werden dabei aufgeteilt. Auch hier erfolgt zuerst die Kennzeichnung der ungültigen Kanten. Anschließend werden Kopien der betroffenen Kanten aufgebrochen und als neue Elemente eingefügt. Die ursprünglichen Linien bleiben im Datenbestand erhalten, sind aber nach dem Speichern aufgrund der Themendefinition nicht mehr sichtbar.

Kante ungültig setzen (Straßengraph)

Nach Auswahl einer Kante enthält das Feld Invalid in der Attributtabelle der Kante das aktuelle Datum und scheint nach dem Speichern aufgrund der Themendefinition nicht mehr im Linienthema auf. Die aktuelle Benutzerkennung wird bei Ausführung des Befehls ebenfalls in der Attributtabelle erfasst.

Straßeninformation (Straßengraph)

Die Maske zur Bearbeitung und Anzeige von Straßendaten wurde bereits im Abschnitt zum Straßenverzeichnis vorgestellt. Ist das Linienthema zur Bearbeitung geöffnet, können die Werte in der Maske verändert und gespeichert werden. Existiert im aktuellen Projekt eine Tabelle mit der Bezeichnung *Strverz.dbf* mit dem numerischen Feld *Strschl* wird diese Tabelle in der Dialogbox angezeigt und der Straßencode kann aus einer Liste ausgewählt werden.



Abbildung 21: Straßeninformationen

8. Die Erfassung des Busliniennetzes der Stadt Steyr

Das Busliniennetz der Stadtbetriebe Steyr wurde als erstes Verwaltungsnetz mit dem neuen Datenmodell erfasst.

Der Hauptgrund für die gewählte Erfassungsart ist die mögliche Mehrfachnutzung der Daten. Einerseits existiert für eine jede Straßenkante des Bezugssystems die Information darüber, ob ein Bus auf einem bestimmten Straßenabschnitt verkehrt, um welche Linie es sich dabei handelt und in welche Richtung sich der Bus dabei bewegt. Zusätzlich gewinnt man eine Datenstruktur, die das neu erfasste Netz als *ungerichteten Graphen* abbildet und somit leicht in einer Sachdatenbank weiterverwendet werden kann.

Die so gewonnenen Informationen sind sowohl untereinander, als auch mit anderen Daten auf dem selben Bezugssystem vergleich- und kombinierbar.

Ein mögliches Beispiel bei Vorliegen eines Radwegenetzes wäre die Abfrage aller Straßenabschnitte an denen Buslinien neben den Radrouten verlaufen.

8.1. Methodik beim Aufbau der Datenbasis

Für die Erfassungsarbeiten standen analoge Unterlagen über die Lage der Haltestellen und den Verlauf der Buslinien zur Verfügung, die freundlicherweise von Stadtbetrieben Steyr bereitgestellt wurden.

Mit den in Abschnitt 7 beschriebenen Werkzeugen für die dynamische Segmentierung konnten die Haltestellen als Bezugspunkte für die dazwischenliegenden Linienabschnitte abgebildet werden. Anschließend wurden, ausgehend vom zentralen Verkehrsknoten (Haltestelle Bahnhof), alle Buslinien in Form von Routen aufgenommen. Die Erfassung erfolgte in der Reihenfolge des Linienvlaufes, jeweils von Haltestelle zu Haltestelle. So kann die Routentabelle als ungerichteter Graph und die Haltestellentabelle als Knotenliste genutzt werden, ohne die Datenstruktur ändern zu müssen. Die Segmenttabelle dient in diesem Zusammenhang derzeit lediglich zur Darstellung der Buslinie auf dem Straßengraphen, da hier die Kantennummern und Stationierungswerte enthalten sind. Zukünftig besteht aber die Möglichkeit diese Tabelle in einer Datenbank, gemeinsam mit anderen Straßeninfrastrukturdaten zu nutzen.

8.2. Kartographische Aufbereitung der Netzplandarstellung

Verkehren mehrere Buslinien auf ein und derselben Straßenkante, so überlagern sich dort verschiedene Routen. Bei der Plandarstellung des Netzes wurde deshalb versucht die Buslinien mit unterschiedlichem Abstand von der Bezugslinie des Straßengraphen darzustellen, und so die Lesbarkeit zu verbessern. ArcView bietet hier die Möglichkeit stationierte Werte auch versetzt darzustellen zu lassen. Die segmentartigen Linienelemente kreuzen sich dann aber an den Knoten der Bezugslinie, oder bilden dort unschöne Lücken.

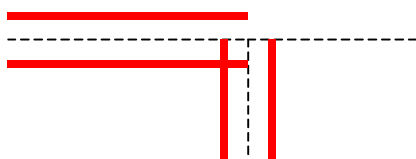


Abbildung 1: Linienversatz in ArcView

Eine derartige Darstellung kann nur vermieden werden, wenn für jede der Buslinien eine eigene Bezugslinie existiert, die auch an den Knoten ohne Unterbrechung durchläuft. Mit

den in diversen CAD-Programmen existierenden Bereinigungsfunktionen für Netzwerke war diese Aufgabe nicht lösbar. Derartige Hilfsmittel zielen im Normalfall auf Knoten-Kanten-Strukturen ab, wodurch es nicht möglich ist Segmente mit gleichartigen Attributwerten automatisch so zusammenzufassen, dass diese auch an den Knoten durchlaufen. Ebenso wenig können damit Linienzüge gebildet werden, die erst in eine Richtung führen, und dann den gleichen Weg wieder zurück nehmen. Die nachstehende Abbildung soll diesen Umstand noch einmal deutlich machen.

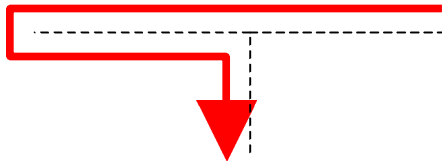


Abbildung 22: Abbiegen nach dem Wenden

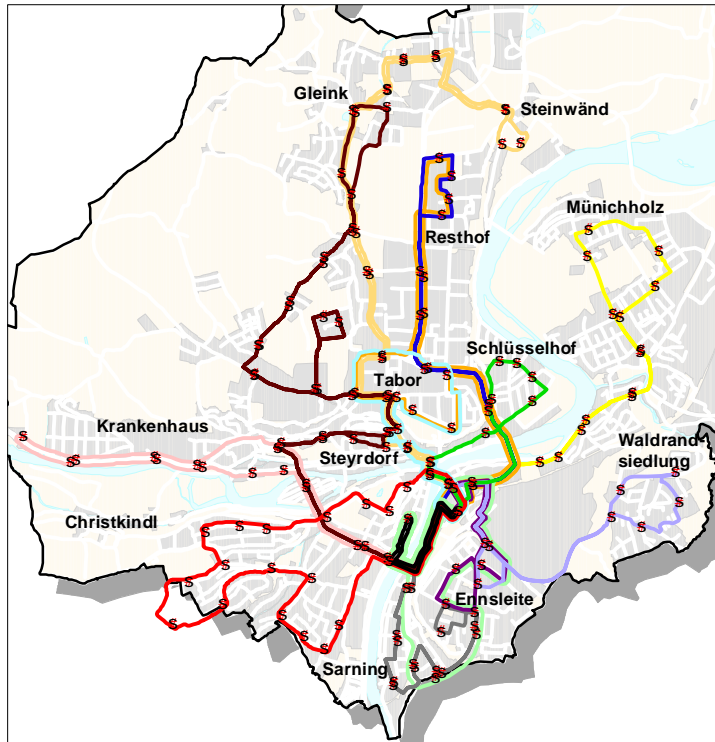
Die abgebildete Zieldarstellungsform war schließlich mit einem kurzen Avenuescript und einer Erweiterung von der Esri-Homepage realisierbar:



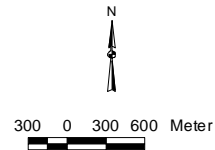
Abbildung 23: Zieldarstellungsform

Um Änderungen an der Geometrie vornehmen zu können waren zuerst die dynamisch segmentierten Linien in eine Shapedatei vom Typ *Polyline* umzuwandeln. Anschließend erfolgte eine Orientierungskorrektur der negativ gerichteten Kanten. Dadurch, dass jene Elemente der Segmenttabelle, deren Bezugskanten entgegen dem Routenverlauf orientiert sind, im Feld *Direction* den Wert -1 enthalten, war eine problemlose Auswahl der negativ gerichteten Segmente möglich. Die Einzelsegmente konnten anschließend richtig gereiht in eine Punkttabelle umgewandelt werden. Das hierzu erstellte Avenuescript *Segment2Point* ist im Anhang B enthalten. Ein Attribut mit der Bezeichnung der Buslinie diente als Ordnungselement in der erzeugten Punktliste. Die abschließende Bildung einer durchgehenden Bezugslinie aus den Daten der Punkttabelle konnte mit Hilfe der User-Extension *Points to Lines or Polylines*¹⁸ erreicht werden. Die Abbildung 24 zeigt die verkleinerte Darstellung des Netzplanes der städtischen Buslinien. Die in der Linienstatistik angegebenen Routenlängen stammen ebenso wie die Anzahl der Haltestellen aus den Tabellen des Verwaltungsnetzes.

¹⁸ D.Patterson, pt2ln2pl.avx (3.11.2000), <http://gis.esri.com/arcsripts>



Liniennetz Stadtbus Steyr



- S Haltestellen
- Buslinien
- 1 Münchenholz
- 10 Steyrdorf-Steinwändweg
- 2/4 Resthof-Tabor
- 2a Resthof
- 2b Krankenhaus
- 3 Stadtplatz-Ennsleite
- 3a Neuschönau-Ennsleite
- 3b Ennsleite
- 4 Steyrdorf-Tabor
- 5 Wieserfeldplatz-Gleink
- 6 Waldrandsiedlung
- 7 Schlüsselhofsiedlung
- 8 Christkindsiedlung
- 9 Stadtplatz

Stand, 3.1.2001

Abbildung 24: Netzplan

Liniennstatistik:

Linie		Länge in km	Haltestellen
1	Bahnhof-Münchenholz	7,63	18
10	Bahnhof-Steyrdorf-Steinwändweg	12,81	34
1a	Bahnhof-BMW-Münchenholz(Werkskurs)	8,30	19
2/4	Bahnhof-Resthof-Tabor	9,60	21
2a	Bahnhof-Resthof	7,85	16
2b	Bahnhof-Krankenhaus	8,58	17
3	Bahnhof-Stadtplatz-Ennsleite	8,35	14
3a	Bahnhof-Neuschönau-Ennsleite	7,44	16
3b	Bahnhof-Ennsleite	4,00	11
4	Bahnhof-Steyrdorf-Tabor	5,07	19
5	Bahnhof-Wieserfeldplatz-Gleink	17,96	40
6	Bahnhof-Waldrandsiedlung	7,02	12
7	Bahnhof-Schlüsselhofsiedlung	3,65	11
8	Bahnhof-Christkindsiedlung	7,81	21
9	Bahnhof-Stadtplatz	2,58	2

9. Projektrückschau und Ausblick

Ursprünglich war vorgesehen auch ein Verwaltungsnetz nach den Richtlinien der RVS 5.011 aufzubauen. Bedingt durch die aufwendige Schlüsselbildung bei der Bezugsliniendefinition in Verbindung mit der Länge und Komplexität des Stadtstraßennetzes, war dieses Vorhaben im Rahmen der vorliegenden Projektarbeit nicht realisierbar. Die unterstützenden Befehle und Eingabehilfsmittel für die Erfassung und den Austausch von Straßendaten nach dieser Norm fehlen derzeit noch. Der Grundstein für den weiteren Ausbau des Straßeninformationssystems konnte mit den im Rahmen dieser Projektarbeit entwickelten Konzepten und Hilfsmitteln aber sehr wohl gelegt werden.

Bei der präsentierten SIS-Erweiterung für ArcView handelt es sich derzeit noch um einen Prototypen, der aufgrund der Erfahrungen bei der zukünftigen Datenerfassung noch zu optimieren zu sein wird. Nachteilig wirkt sich dabei aus, dass der GIS-Hersteller ESRI die Produktlinien ArcView und ArcInfo zusammenführt und die Programmiersprache Avenue in den neuen Versionen von ArcGIS nicht mehr unterstützt wird. Das Shape-Dateiformat für GIS-Daten bleibt dort aber ebenso wie das Modell der dynamischen Segmentierung weiterhin nutzbar.

Die Übernahme der Daten in ein Fremdsystem wird momentan als unproblematisch eingeschätzt. Die einfache geometrische Struktur des als Bezugssystem dienenden Straßengraphen kann mit geringem Programmieraufwand in eine einfache Punkteliste umgewandelt werden. Die anderen Informationen liegen sämtlich in Form von dBASE-Tabellen vor und können somit leicht in herkömmliche Datenbankumgebungen eingebunden werden.

10. Literatur

- DUTTON, G. (1992):** Handling Positional Uncertainty in Spatial Databases. In: Proceedings SHD'92: 460-469
- ESRI (1996) I:** Avenue: Customization and Application Development for ArcView GIS (Benutzerhandbuch des Softwareherstellers ESRI), Redlands CA, 1996.
- ESRI (1996) II:** ArcView Network Analyst: Optimum Routing, Closest Facility and Service Area Analysis (Benutzerhandbuch des Softwareherstellers ESRI), Redlands CA, 1996.
- GOBIET, W (2000):** Datenmodell für ein einheitliches Bezugssystem für Verkehrs- und Verkehrswegedatenbanken in Österreich. In: ZAGEL, B. (Hrsg.): GIS in Verkehr und Transport, Heidelberg 2000
- GUPTILL, S.C. und MORRISON, J.L., (1995):** Elements of spatial data quality. Oxford: Elsevier Science Ltd.
- KOLLARITS S (1999):** Verkehr ist mehr. Modellgrundlagen zur (Re-)Integration von Verkehrsinfrastruktur, Verkehrsplanung und Verkehrsmanagement. In: SCHRENK M. (Hrsg.): CORP 99 Beiträge zum 4. Symposium zur Rolle der Informationstechnologie in der Raumplanung , Wien 1999
- MAURER, P (2000):** Visualisierung des Unfallgeschehens in einem Geographischen Informationssystem (GIS). . In: ZAGEL, B. (Hrsg.): GIS in Verkehr und Transport, Heidelberg 2000
- PONN, A. (1996):** Salzburger Straßenverkehrsinformationssystem. In: DOLLINGER,F. und J.STROBL: Angewandte Geograph. Informationsverarbeitung VIII = Salzburger Geographische Materialien, Heft 24.
- RAZAVI A.H. und WARWICK V.(1997):** ArcView Gis/Avenue Programmer's Reference, OnWord Press, Camino Entrada
- RAZAVI A.H. (1997):** ArcView Gis/Developer's Guide, OnWord Press, Camino Entrada

ANHANG A

Straßenkarte Steyr

Anhang B

Avenuescripts

Bezeichnung	Seite
Sis.AddUserData.....	1
Sis.BP.lbt_Abrbruch.Click.....	2
Sis.BPOpen.....	2
Sis.BP.txtl_Stat.Changed.....	2
Sis.BPMkTable.....	3
Sis.BPSet.....	3
Sis.CalcNodes.....	6
Sis.FindEdge.....	9
Sis.FindIntersection.....	10
Sis.FindOpenNode.....	11
Sis.FindOverlay.....	13
Sis.FindPLerror.....	14
Sis.FlipSelected.....	17
Sis.GetUserTheme.....	18
Sis.IsLine.Update.....	19
Sis.IsPoint.Update.....	20
Sis.Kt.Lbt_Abrbruch.Click.....	20
Sis.Kt.Lbt_OK.Click.....	20
SIS.Kt.lbx_StrVerz.Select.....	21
SIS.Kt.Open.....	21
SIS.Kt.Recalc.....	23
SIS.Kt.Set.....	24
Sis.KtUnion.....	26
SIS.ListSub.....	28
SIS.LnDefine.Read.....	28
SIS.LnDefine.Write.....	29
SIS.MeasurePolyline.....	30
SIS.MkFTheme.....	32
SIS.MkRandomPoint.....	33
SIS.MkTable.....	34
SIS.MkTurnTable.....	35
SIS.MPtoP.....	38
SIS.PLineM.List.....	40
SIS.PtToLineTbl.....	40
SIS.ReturnLSegAzimuth.....	43
SIS.RtDef.....	44
SIS.RtRVSDef.....	51
SIS.RtMkTables.....	57
SIS.RtRemove.....	59
SIS.SetUser.....	61
SIS.ShowStreet.....	61
SIS.Split.....	63
SIS.SplitUpdate.....	65
SIS.StrGraphMake.....	65
SIS.StrInvalid.....	66
SIS.StrInvalidByPoint.....	67
SIS.StrVerzMkTable.....	68
SIS.TabIndex.....	68
SIS.Table.List.....	69

SIS.Table.SortPhysical	70
SIS.TableDoc	74
SIS.TableReportFile	75
SIS.TrnTblThemeMake	76
SIS.ViewDP.List	77
SISExt.Install	77
SISExt.Make	78
SISExt.UnInstall	79
SISExt.Unload	80
Segment2Point	80
SIS.RtToPolyline	81
SIS.ViewDPln.List	83

```

' Name:   Sis.AddUserData
'
' Title:  Benutzererkennung und Datum an FTab anfügen
'
' Topics: SIS, Tools
'
' Description: Hängt an die Ftab die Felder User und Date an, bzw. wird,
'              wenn diese Felder vorhanden sind der User und das Datum
aktualisiert
'
'
' Requires: View mit mindestens einem Thema
'
' Self:
'
' Returns:

t = "User/Date"

theView = av.GetActiveDoc

theTheme = theView.GetActiveThemes.Get(0)

if((theTheme.hasTable) and (theTheme.Is(FTheme) and
(not(theTheme.Is(Library))))))then

    theFtab = theTheme.GetFtab

    theUserField = theFtab.FindField("User")
    theDateField = theFtab.FindField("Date")

    theFtab.SetEditable(true)

    if(theFtab.IsEditable)then
        if(theUserField = nil) then
            theUserField = Field.Make("User", #FIELD_CHAR, 10, 0)
            theFtab.AddFields({theUserField})
        end

        if(theDateField = nil) then
            theDateField = Field.Make("Date", #FIELD_DATE, 10, 0)
            theFtab.AddFields({theDateField})
        end

        theDate = Date.Now

        if(theFtab.getSelection.Count = 0)then
            for each rec in theFtab
                theFtab.SetValue(theUserField, rec, _theUser)
                theFtab.SetValue(theDateField, rec, theDate)
                av.ShowMsg("Aktualisiere Datensatz"++rec.AsString)
            end
        else
            for each rec in theFtab.getSelection
                theFtab.SetValue(theUserField, rec, _theUser)
                theFtab.SetValue(theDateField, rec,
                    theDate.SetFormat("yyyy-mm-dd"))
                av.ShowMsg("Aktualisiere Datensatz"++rec.AsString)
            end
        end

        theFtab.SetEditable(false)
    else
        MsgBox.Error("Diese Thema kann nicht ergänzt werden!",t)

```

```

        exit
    end
else
    MsgBox.Error("Diese Thema kann nicht ergänzt werden!",t)
end

'-----Scriptende-----

'Sis.BP.lbt_Abbruch.Click
'Teil des Abbruchbuttons von lbt_Abbruch
'Schließt den Dialog ohne Speichern
'des BP's
self.GetDialog.SetModalResult(0)
self.GetDialog.Close
'Sis.BP.lbt_OK.Click
'Teil des OK-Buttons von lbt_OK
'Beendet den Dialog und speichert
'die Werte
self.GetDialog.SetModalResult(1)
self.GetDialog.Close

'-----Scriptende-----

'SIS.BPOpen
'Öffnet den Dialog SisBP und ordnet den
'txl-Feldern die Werte des Bezugspunktes zu

RouteID = self.Get(0)
Rlen = self.Get(1)
Stat = self.Get(2)
Offs = self.get(3)

theDialog = av.FindDialog("SISBP")
theDialog.FindByName("txl_RouteID").SetText(RouteID)
theDialog.FindByName("txl_Rlen").SetText(Rlen)
theDialog.FindByName("txl_Stat").SetText(Stat)
theDialog.FindByName("txl_Offs").SetText(Offs)
theDialog.FindByName("txl_BPInfo").SetText("0")
ret = theDialog.Open
if(ret = 1) then
    return({theDialog.FindByName("txl_Stat").GetText.AsNumber,
            theDialog.FindByName("txl_Offs").GetText.AsNumber,
            theDialog.FindByName("txl_BPInfo").GetText})
else
    return(nil)
end

'-----Scriptende-----

'SIS.BP.txl_Stat.Changed
'Testet ob der eingegebene Wert kleiner
'ist als die Gesamtlänge der Kante

Rlen = self.GetDialog.FindByName("txl_Rlen").GetText.AsNumber
MyStat = self.Gettext.AsNumber

if((MyStat > Rlen)or(MyStat < 0)) then
    MsgBox.Error("Der Stationierungswert muss kleiner sein als die
    Kantenlänge und größer 0!","SIS-Error")
    self.Settext(Rlen.AsString)
end

```

```
'-----Scriptende-----

' Name:   SIS.BPMkTable
'
' Zweck:  Erstellen einer für ein Bezugspunktthema erforderlichen
'         Tabelle
'
'
' Thema:  SIS, Dynamische Sementierung
'
' Voraussetzung: SIS.MkTable wird aufgerufen
'
'
' Self:
'
' Rückgabe:
'
'
' Autor:  Horst Baumgartner 2001/11/21
'
'
' Felder für die Bezugspunkttable
' PID     PunktID (Schlüssel)
' Stat    Stationierung am Straßengraphen
' Offset  Ordinate am Straßengraphen
' BPInfo  Textfeld für Benutzereingabe
' Rot     Rotationswinkel für das Symbol
' User    Benutzererkennung
' Date    Datum

theFieldList = {
Field.Make("Pid",#FIELD_DOUBLE,16,0),
Field.Make("Stat",#FIELD_DOUBLE,16,2),
Field.Make("Offset",#FIELD_DOUBLE,16,2),
Field.Make("Str_id",#FIELD_DOUBLE,16,0),
Field.Make("Rot",#FIELD_DOUBLE,10,4),
Field.Make("Info",#FIELD_CHAR,30,0),
Field.Make("X",#FIELD_DOUBLE,16,2),
Field.Make("Y",#FIELD_DOUBLE,16,2),
Field.Make("User",#FIELD_CHAR,10,0),
Field.Make("Date",#FIELD_DATE,8,0)
}

av.Run("SIS.MkTable",{ "Bezugspunkttable
anlegen", "UBPunkt.dbf",theFieldList,false})

'-----Scriptende-----

' Name:   SIS.BPSet
'
' Zweck:  Das aktive Thema wird nach dem Segmentierungsthema durchsucht,
'         und ein Bezugspunkt kann durch Eingabe einer Linie auf einem
'         Straßenabschnitt stationiert werden.
'
'
' Thema:  SIS, Dynamische Sementierung
'
' Voraussetzung: Das aktive Dokument muss ein View sein. Das Aktive Thema
muss
'
'           ein Punktthema aus einer dynamischen Segmentierung sein.
Ist
```

```

'           durch 'SIS.ISPoint.Update' im Button sicherzustellen.
'
' Self:
'
' Rückgabe:
'
' Autor: Horst Baumgartner 2000/02/13
'

theView = av.GetActiveDoc
theDisplay = theView.GetDisplay

' Dyn.Punktthema muss das aktive Thema sein

theBPTheme = theView.GetActiveThemes.Get(0)
theSrcName = theBPTheme.GetSrcName

' Die Ftab des Routenthemas holen

theFtab = theSrcName.GetRouteSys

' Die Route mittels einer Linie gleichzeitig selektieren
' und den Bezugspunkt an den Anfang der definierten Linie setzen

theSLine = theDisplay.ReturnUserLine
theFtab.SelectByLine(theSLine, #VTAB_SELTYPE_NEW)

thePoint = theSLine.ReturnStart

shapeField = theFtab.FindField("Shape")

' Das Routenfeld vom Linienthema holen

IDField = theSrcName.GetRatRouteId

' Selektion auswerten

theBitmap = theFtab.GetSelection

'nur wenn auch ein Element ausgewählt wurde

if(theBitmap.Count = 0) then
    exit
end

rec = theBitmap.getNextSet(-1)

' Polyline shape abfragen

thePolyline = theFtab.ReturnValue(shapeField,rec)
StrID = theFtab.ReturnValue(IDField,rec)

Plen = thePolyline.ReturnLength

' Distanz zum Punkt ermitteln
    theDistance = 0
    theDistance = thePolyline.QueryPointDistance(thePoint,theDistance) *
        (-1)

' Winkel für die Symbolrotation
    Rot =
    av.Run("SIS.ReturnLSegAzimuth", {thePoint,thePolyline.Along(thePolyline.P
ointPosition(thePoint))})

' Stationierung von Prozent auf Länge umrechnen

    Stat = thePolyline.PointPosition(thePoint) * Plen / 100.0

```

```

' Info für den Bezugspunkt eingeben oder editieren
' Result ist nil wenn die Routine abgebrochen wurde

result =
av.Run("SIS.BP.Open",{StrID.AsString,Plen.AsString,Stat.AsString,theDistanc
e.AsString})

if (result = nil) then
  Exit
end

' Werte aktualisieren

Stat = result.Get(0)
theDistance = result.Get(1)
BPText = result.Get(2)

' Stationierungstabelle befüllen

theVTab = theSrcName.GetEventTable

' Routen Felder holen

BPStr_ID = TheSrcName.GetEventRouteId
BPStat = TheSrcName.GetMeasureField
BPOffset = TheSrcName.GetOffsetField

' Punkttabelle zum Editieren öffnen

theVTab.StartEditingWithRecovery

' Felder für Zähler und Info suchen und allenfalls anlegen
' BPPid dient als eindeutiger Zähler und muss erzeugt werden
' BPInfo dient zur ergänzenden Beschreibung des Punktes

BPPid = theVTab.FindField("Pid")

if(BPPid = nil) then
  BPPid = Field.Make("Pid",#FIELD_DOUBLE,16,0)
  theVTab.AddFields({BPPid})
end

BPInfo = theVTab.FindField("Info")
if(BPInfo = nil) then
  BPInfo = Field.Make("Info",#FIELD_CHAR,30,0)
  theVTab.AddFields({BPInfo})
end

BPRot = theVTab.FindField("Rot")
if(BPRot = nil) then
  BPRot = Field.Make("Rot",#FIELD_DOUBLE,10,4)
  theVTab.AddFields({BPRot})
end

BPX = theVTab.FindField("X")
if(BPX = nil) then
  BPX = Field.Make("X",#FIELD_DOUBLE,16,2)
  theVTab.AddFields({BPX})
end

BPY = theVTab.FindField("Y")
if(BPY = nil) then
  BPY = Field.Make("Y",#FIELD_DOUBLE,16,2)

```

```

        theVTab.AddFields({BPY})
end

BPUser = theVTab.FindField("User")
if(BPUser = nil) then
    BPUser = Field.Make("User",#FIELD_CHAR,10,0)
    theVTab.AddFields({BPUser})
end

BPDate = theVTab.FindField("Date")
if(BPDate = nil) then
    BPDate = Field.Make("Date",#FIELD_DATE,8,0)
    theVTab.AddFields({BPDate})
end

' nächste freie Nummer holen

bpix = av.Run("SIS.TabIndex",{theVtab,BPPid})

rec = theVTab.AddRecord
theVTab.SetValue(BPStr_ID, rec, StrID)
theVTab.SetValue(BPPid, rec, bpix)
theVTab.SetValue(BPOffset, rec, theDistance)
theVTab.SetValue(BPStat, rec, Stat)
theVTab.SetValue(BPInfo, rec, BPText)
theVTab.SetValue(BPRot,rec, Rot)
theVTab.SetValue(BPX,rec, thePoint.GetX)
theVTab.SetValue(BPY,rec, thePoint.GetY)
theVTab.SetValue(BPUser,rec, _theUser)
theVTab.SetValue(BPDate,rec, Date.Now)

theVTab.StopEditingWithRecovery(True)

av.GetProject.SetModified(true)

'-----Scriptende-----

' Name: SIS.CalcNodes
'
'
' Title: Renumbers nodes for arc features and and update ( or create if
not exist ) values for
'         FNODE and TNODE in the active theme. Active theme must be
polyline.
'
'
' Topics: FNODE, TNODE, Hydraulic model,CYBER NET
'
' Description: Create/Update field FromNode and ToNode in the active
themes.
'
' Requires: A View must be the active document.
'
' Self:
'
' Returns:
' Author : original script - Roland Salgaro - Hillsborough County Water
Department

```

```

'          modify - Andrej TRDIN - Water supplay system, Velenje, SLOVENIA
'          modified Sep. 6, 2000 - Gord Haines, Lheidli T'enneh First
Nation, Prince George, B.C., Canada
'          - added node valence to the output point
theme, compacted code
'          - ISLAND POLYGONS MADE UP OF ONLY ONE LINE
WILL HAVE NODE VALENCE OF 2!
'          - SEE BELOW IF YOU WANT THIS CHANGED
'
' View should be the Active Document

theView = av.GetActiveDoc
'
' Get the list of active themes. if there aren't any, let the user know
' and exit.
'
theActivethemeList = theView.GetActivethemes
if (theActivethemeList.Count = 0) then
    MsgBox.Error("No active themes.", "")
    Exit
end

' List to hold node locations:
    PointList = List.Make

' List to hold node valence - G.H.
    ValenceList = List.Make

t = theView.GetActiveThemes.Get(0)

theFTab      = t.GetFTab
shapeField   = theFTab.FindField( "Shape" )
numRecs      = theFTab.GetNumRecords

if (shapeField.GetType <> #FIELD_SHAPELINE) then
    MsgBox.Info("Theme must be a POLYLINE.", "STOP ")
    return(nil)
end

theFTab.SetEditable(true)

' if it's polyline check for the existence of the fields "FromNode" and
' "ToNode". if they do not exist, create them.

if (theFTab.FindField("FromNode") = nil) then
    theFromNodeField = Field.Make("FromNode", #FIELD_DOUBLE, 12, 0)
    theFTab.AddFields({theFromNodeField})
else
    theFromNodeField = theFTab.FindField("FromNode")
end

if (theFTab.FindField("ToNode") = nil) then
    theToNodeField = Field.Make("ToNode", #FIELD_DOUBLE, 12, 0)
    theFTab.AddFields({theToNodeField})
else
    theToNodeField = theFTab.FindField("ToNode")
end

av.ShowStopButton
av.ShowMsg( "Exporting"++t.GetName+"..." )

for each recNum in theFTab
    currentShape = theFTab.ReturnValue( shapeField, recNum )
    shapeList = currentShape.AsList

    for each shapePart in shapeList

```

```

for each x in {0,(shapePart.Count-1)}
  thePnt = shapePart.Get(x)
  pntCheck = PointList.FindByValue(thePnt)

  if (pntCheck > -1) then '*thePnt was found in PointList
    pntValence = ValenceList.Get(pntCheck) + 1
    ValenceList.Set(pntCheck,pntValence) '*COMMENT IF YOU WANT ISLAND
    POLYGONS WITH VALENCE OF 1!
    if (x = 0) then '*From node
      FromNode = pntCheck
      '*ValenceList.Set(pntCheck,pntValence) '*COMMENT IF YOU WANT
      ISLAND POLYGONS WITH VALENCE OF 2!
    else
      ToNode = pntCheck
      '*if (thePnt <> shapePart.Get(0)) then '*Make sure the from node
      and to node aren't the same
      '*ValenceList.Set(pntCheck,pntValence) '*COMMENT IF YOU WANT
      ISLAND POLYGONS WITH VALENCE OF 2!
      '*end
    end '*end if (x

  else '*thePnt is a new one
    if (x = 0) then '*From node
      FromNode = PointList.Count
    else
      ToNode = PointList.Count
    end '*end if (x

    PointList.Add(thePnt)
    ValenceList.Add(1)

  end '*end if (pntCheck
end '*end for each x

' Add values to theFtab:
theFtab.SetValue(theFromNodeField,recNum,FromNode)
theFtab.SetValue(theToNodefield,recNum,ToNode)
end '*end for each shapePart

progress = (recNum / numRecs) * 100
proceed = av.SetStatus( progress )
if ( proceed.Not ) then
  av.ClearStatus
  av.ShowMsg( "Stopped" )
  exit
end

end '*end for each recNum

' Close the FTab to editing:
theFtab.SetEditable(false)

av.ClearStatus
av.ClearMsg

' Create a new shape file for the nodes created:
defaultName = FileName.Make("$Home").MakeTmp("Node","shp")
shpName = FileDialog.Put( defaultName,"*","Output NODE theme")

if (shpName = nil) then
  exit
end

shpName.SetExtension("shp")

' Make the shape file:

```

```

shpFTab = FTab.MakeNew(shpName,Point)

' Set up fields for the FTab:
fields = List.Make
fields.Add(Field.Make("Node_ID",#FIELD_LONG,12,0))
fields.Add(Field.Make("Valence",#FIELD_LONG,12,0))
fields.Add(Field.make("User",#FIELD_CHAR,10,0))
fields.Add(Field.make("Date",#FIELD_DATE,8,0))
shpFTab.AddFields(fields)

shpField = shpFTab.FindField("Shape")
idField = shpFTab.FindField("Node_ID")
valField = shpFTab.FindField("Valence")
UserField = shpFTab.FindField("User")
DateField = shpFTab.FindField("Date")

av.ShowStopButton
av.ShowMsg("Creating nodes - Shape File")

Nodes = PointList.Count - 1
for each i in 0..Nodes
  thePoint = PointList.Get(i)
  rec = shpFTab.AddRecord
  shpFTab.SetValueNumber(idField,rec,i)
  shpFTab.SetValueNumber(valField,rec,(ValenceList.Get(i)))
  shpFTab.SetValue(shpField,rec,thePoint)
  shpFTab.SetValue(UserField,rec,_theUser)
  shpFTab.SetValue(DateField,rec,Date.Now)

  progress = (i/Nodes) * 100
  proceed = av.SetStatus( progress )
  if (proceed.Not) then
    av.ClearStatus
    av.ShowMsg("Stopped")
    break
  end
end
shpFTab.Flush

NewTheme = Theme.Make(SrcName.Make(shpName.AsString))
theView.AddTheme(NewTheme)
av.ClearStatus
av.ClearMsg

av.ShowMsg( shpFTab.GetNumRecords.AsString++" nodes created")

'-----Scriptende-----

' Name: SIS.FindEdge
'
' Zweck: Suche nach einer bestimmten Kante in einer Kantenliste
'
'
' Thema: SIS, Dynamische Sementierung
'
' Voraussetzung: Wird von SIS.RtDef aufgerufen
'
' Self: Liste mit
'         0 Kantenliste
'         1 StraßenStartId
'         2 FromNode
'         3 StraßenEndID
'
' Rückgabe: Listenindex wenn die Kante gefunden wurde, sonst nil

```

```

'
' Autor: Horst Baumgartner 2002/01/01
'
'

theList = self.get(0)
theSID = self.get(1)
theFrom = self.get(2)
theEID = self.get(3)

for each elem in 0 .. (theList.Count-1)
  if((theList.get(elem).get(0) <> theSID) and
    (theList.get(elem).get(1) = theFrom) and
    (theList.get(elem).get(0) <> theEID)) then
    return elem
  end
end
return nil

'-----Scriptende-----

' Name:  Sis.FindIntersection
'
' Title:  Suchen aller Schnittpunkte ohne Knoten
'
' Topics:  SIS, Topologieprüfung
'
' Description:  Durchsucht ein Thema vom Typ Polyline
'               und markiert alle gefundenen Schnittpunkte
'
'
' Requires:  View mit einem Linienthema
'
' Self:
'
' Returns:

theCnt = 0
theView = av.GetActiveDoc
theTheme = theView.GetActiveThemes.get(0)
theFtab = theTheme.getFtab

if(theFtab.GetShapeClass.GetClassName <> "Polyline") then
  MsgBox.Error("Das aktive Thema muss vom Typ Polyline sein.,"SIS")
  exit
end

' Punkteliste für die ermittelten Schnittpunkte

theSPList = {}

theShapeField = theFtab.FindField("Shape")

theRecAnz = theFtab.GetNumRecords

for each recl in theFtab
  thefShape = theFtab.ReturnValue(theShapeField,recl)

  sp = thefShape.Along(0)
  ep = thefShape.Along(100)

  av.ShowMsg("Suche Kreuzungen...")
  av.SetStatus(recl / (theRecAnz / 100))

  theTheme.SelectByPolyline(thefShape,#VTAB_SELTYPE_NEW)

```

```

theBitmap = theFtab.GetSelection

for each rec2 in theBitmap
  theShape = theFtab.ReturnValue(theShapeField,rec2)

  if(theShape.Intersects(theFShape) = true) then
    pt = theShape.PointIntersection(theFShape)

    ' pt ist multipoint shape

    thePtlist = pt.AsList

    for each spt in thePtList
      if((spt = sp).not and (spt = ep).not) then
        ' Wenn noch nicht vorhanden
        if(theSPList.FindbyValue(spt) = -1) then
          theSPList.Add(spt)
        end
      end
    end
  end
end
end
end

theBitmap.ClearAll

av.ClearStatus
av.ClearMsg

for each spt in theSPList
  gp = GraphicShape.Make(spt)

  'Farbe und Größe der Punkte festlegen

  theSymbol = gp.GetSymbol
  theSymbol.SetColor(Color.GetRed)
  theSymbol.SetSize(7)

  theView.GetGraphics.Add(gp)
end

theView.GetGraphics.UnselectAll
av.GetProject.SetModified(true)
av.ProcessAllInvals

msgBox.Info("Anzahl der Schnittpunkte:"++ theSPList.Count.AsString,"SIS")

'----Scriptende-----

' Name:  Sis.FindOpenNode
'
' Title:  Suchen aller Knoten innerhalb eines angegebenen Schwellenwertes
'
' Topics:  SIS, Topologieprüfung
'
' Description:  Durchsucht ein Knotenthema vom Typ Point
'                und markiert jene Punkte die näher als der Schwellenwert
'                zusammenliegen
'
' Requires:  View mit einem Punktthema
'
' Self:
'
' Returns:
theView = av.GetActiveDoc

```

```

theTheme = theView.GetActiveThemes.get(0)
theFtab = theTheme.getFtab

if(theFtab.GetShapeClass.GetClassName <> "Point") then
  MsgBox.Error("Das aktive Thema muss vom Typ Point sein.,"SIS")
  exit
end

' Punkteliste für die ermittelten offenen Knoten

theOPList = {}

theSnapStr = msgBox.Input("Lagetoleranz:","Offene Knoten","0.01")

if(theSnapStr = nil) then
  exit
else
  theSnapNum = theSnapStr.AsNumber
end

theShapeField = theFtab.FindField("Shape")

theRecAnz = theFtab.GetNumRecords

for each rec1 in theFtab
  thefShape = theFtab.ReturnValue(theShapeField,rec1)
  rec2 = rec1 + 1

  av.ShowMsg("Suche offene Knoten...")
  av.SetStatus(rec1 / (theRecAnz / 100))

  theTheme.SelectByPolygon(Circle.Make(thefShape,
theSnapNum).AsPolygon,#VTAB_SELTYPE_NEW)
  theBitmap = theFtab.GetSelection

  for each rec2 in theBitmap
    thesShape = theFtab.ReturnValue(theShapeField,rec2)

    if((thesShape.Distance(thefShape) < theSnapNum) and (thesShape <>
theFShape)) then
      gp1 = theFShape
      gp2 = thesShape
      if(theOPList.FindbyValue(gp1) = -1) then
        theOPList.Add(gp1)
      end
      if(theOPList.FindbyValue(gp2) = -1) then
        theOPList.Add(gp2)
      end
    end
  end
end

theBitmap.ClearAll

av.ClearStatus
av.ClearMsg

for each op in theOPList
  gp = GraphicShape.Make(op)

  'Farbe und Größe der Punkte festlegen

  theSymbol = gp.GetSymbol
  theSymbol.SetColor(Color.GetGreen)
  theSymbol.SetSize(7)

```

```

    theView.GetGraphics.Add(gp)
end

theView.GetGraphics.UnselectAll
av.GetProject.SetModified(true)

av.ProcessAllInvals

msgBox.Info("Anzahl der offenen Knoten im Abstand
von"++theSnapStr++"Einheiten:"++ theOPList.Count.AsString,"SIS")

'----Scriptende-----

' Name:  Sis.FindOverlay
'
' Title:  Suchen aller Überlagerungen von Polylinien im aktuellen Thema
'
' Topics:  SIS, Topologieprüfung
'
' Description:  Durchsucht ein Thema vom Typ Polyline
'               und markiert alle gefundenen Überlagerungen
'
'
' Requires:  View mit einem Linienthema
'
' Self:
'
' Returns:
'

theView = av.GetActiveDoc
theTheme = theView.GetActiveThemes.get(0)
theFtab = theTheme.getFtab

if(theFtab.GetShapeClass.GetClassName <> "Polyline") then
  MsgBox.Error("Das aktive Thema muss vom Typ Polyline sein.,"SIS")
  exit
end

' Liste mit Überlagerungen

thePlnList = {}

theShapeField = theFtab.FindField("Shape")

theRecAnz = theFtab.GetNumRecords

for each rec1 in theFtab
  thefShape = theFtab.ReturnValue(theShapeField,rec1)
  rec2 = rec1 + 1

  av.ShowMsg("Suche Überlagerungen...")
  av.SetStatus(rec1 / (theRecAnz / 100))

  theTheme.SelectByPolyline(thefShape,#VTAB_SELTYPE_NEW)
  theBitmap = theFtab.GetSelection

  for each rec2 in theBitmap
    thesShape = theFtab.ReturnValue(theShapeField,rec2)

    if(thesShape.Intersects(thefShape) = true) then
      pl = thesShape.LineIntersection(thefShape)

      ' nur wenn es sich nicht um eine Überlagerung mit sich selbst

```

```

        handelt

        if((pl <> thefShape) and (pl.ReturnLength > 0.001)) then
            if(thePlnList.FindbyValue(pl) = -1) then
                thePlnList.Add(pl)
            end
        end
    end
end
end
end

theBitmap.ClearAll

' Fehlerfiguren generieren

for each pl in thePlnList
    gl = GraphicShape.Make(pl)

    'Farbe und Größe der Linie festlegen

    theSymbol = gl.GetSymbol
    theSymbol.SetColor(Color.GetBlue)
    theSymbol.SetSize(2)

    theView.GetGraphics.Add(gl)
end

av.ClearStatus
av.ClearMsg
theView.GetGraphics.UnselectAll
av.GetProject.SetModified(true)
av.ProcessAllInvals

msgBox.Info("Anzahl der Überlagerungen:"++thePlnList.Count.AsString,"SIS")

'-----Scriptende-----

' Name:  Sis.FindPLError
'
' Title:  Suchen aller Kreuzungen und Überlagerungen innerhalb der
Polylinien
'
' Topics:  SIS, Topologieprüfung
'
' Description:  Durchsucht ein Thema vom Typ Polyline
                und markiert alle gefundenen Schnittpunkte und
Überlagerungen
'
' Requires:  View mit einem Linienthema
'
' Self:
'
' Returns:

theCnt = 0
theView = av.GetActiveDoc
theTheme = theView.GetActiveThemes.get(0)
theFtab = theTheme.getFtab

if(theFtab.GetShapeClass.GetClassName <> "Polyline") then
    MsgBox.Error("Das aktive Thema muss vom Typ Polyline sein.,"SIS")
    exit
end
end

```

```

' Punkteliste für die ermittelten Schnittpunkte
theSPList = {}

' Linienliste mit Überlagerungen
thePlnList = {}

' Liste mit den Polylinienstützpunkten
thePCntList = {}

' Liste mit unzusammenhängenden Abschnitten
theXPlnList = {}

theShapeField = theFtab.FindField("Shape")

theRecAnz = theFtab.GetNumRecords

for each recl in theFtab
  thefShape = theFtab.ReturnValue(theShapeField,recl)
  av.ShowMsg("Suche Fehler innerhalb der Polylinien...")
  av.SetStatus(recl / (theRecAnz / 100))

  ' Test ob die Polylinie aus mehreren unzusammenhängenden Teilen besteht
  if(thefShape.CountParts > 1) then
    theXPlnList.Add(thefShape)
  end

  thePlPtList = theFShape.AsMultipoint.AsList

  ' nur wenn die Polylinie aus mehr als einer Kante besteht
  thePtAnz = thePlPtList.Count

  if(thePtAnz > 2) then
    ptlix = 0
    while(ptlix < (thePtAnz - 1))
      sp = thePlPtList.Get(ptlix)
      ep = thePlPtList.Get(ptlix + 1)
      pt2ix = ptlix + 1
      seg1 = PolyLine.Make({{thePlPtList.Get(ptlix),thePlPtList.Get(ptlix +
1)}})
      while(pt2ix < (thePtAnz - 1))
        seg2 = PolyLine.Make({{thePlPtList.Get(pt2ix),thePlPtList.Get(pt2ix
+ 1)}})
        if(seg1.Intersects(seg2) = true) then
          pt = seg1.PointIntersection(seg2)

          ' pt ist multipoint shape

          thePtlist = pt.AsList

          for each spt in thePtList
            if((spt = sp).not and (spt = ep).not) then
              ' Wenn noch nicht vorhanden
              if(theSPList.FindbyValue(spt) = -1) then
                theSPList.Add(spt)
              end
            end ' ende spt = sp
          end ' ende for each

          ' Test ob sich die Segmente überlagern
          pl = seg1.LineIntersection(seg2)

```

```

' Werte unter 0.001 m werden ignoriert
if(pl.ReturnLength > 0.001) then
  if(thePlnList.FindbyValue(pl) = -1) then
    thePlnList.Add(pl)
  end
  end '*Ende ((pl <>
end '* Ende segl.intersects
pt2ix = pt2ix + 1
end ' ende while(pt2ix < thePtAnz - 1)
ptlix = ptlix + 1
end ' ende while(ptlix < thePtAnz)

' es sind auch Kreuzungen über einen der Stützpunkte möglich
' das ist gesondert zu prüfen, der Stützpunkt scheint dann
' mehrfach in der Liste auf

' Liste leeren

thePCntList.Empty

for each tp in thePlPtList
  ' Test ob der Punkt vorhanden ist
  if(thePCntList.FindbyValue(tp) > -1) then
    ' wenn erster Punkt = letzter Punkt, dann OK, sonst Fehler
    if(tp <> thePlPtList.Get(thePtAnz - 1)) then
      theSPList.Add(tp)
    end
  else
    thePCntList.Add(tp)
  end
end '*ende for each tp
end ' ende if thePtAnz > 2
end '*ende for each rec

av.ClearStatus
av.ClearMsg

' Schnittpunkte anzeigen

for each spt in theSPList
  gp = GraphicShape.Make(spt)

  'Farbe und Größe der Punkte festlegen

  theSymbol = gp.GetSymbol
  theSymbol.SetColor(Color.GetRed)
  theSymbol.SetSize(7)

  theView.GetGraphics.Add(gp)
end

' Überlagerungen anzeigen

for each oLn in thePlnList
  gp = GraphicShape.Make(oLn)

  'Farbe und Größe der Punkte festlegen

  theSymbol = gp.GetSymbol
  theSymbol.SetColor(Color.GetBlue)
  theSymbol.SetSize(2)

  theView.GetGraphics.Add(gp)
end

' Mehrfach-Polylinien anzeigen

```

```

for each xLn in theXPlnList
  gp = GraphicShape.Make(xLn)

  'Farbe und Größe der Linie festlegen

  theSymbol = gp.GetSymbol
  theSymbol.SetColor(Color.GetGreen)
  theSymbol.SetSize(2)

  theView.GetGraphics.Add(gp)
end

theView.GetGraphics.UnselectAll
av.GetProject.SetModified(true)
av.ProcessAllInvals

msgBox.Info("Anzahl der Schnittpunkte(rot):"++ theSPList.Count.AsString+ NL
+ "Anzahl der Überlagerungen(blau):"++
    thePlnList.Count.AsString+ NL + "Anzahl
    Mehrfach-Polylinien(grün):"++
    theXPlnList.Count.AsString,"SIS")

'-----Scriptende-----

' Name: SIS.FlipSelected
'
' Title: Polylinienorientierung ändern
'
' Topics: SIS, Bearbeitungsfunktionen, Darstellung
'
' Description: Ändert die Orientierung aller selektierter Elemente
'              in einem Polylinienthema
'
'
' Requires: View mit einem Linienthema
'
' Self:
'
' Returns:
'
' Author: Michael Herter 07.12.1999 mailto:M.Herter@esri-germany.de
'         www.esri.com/arcs-scripts
'

theView = av.GetActiveDoc
theThemes = theView.GetActiveThemes
if (theThemes.Count = 0) then
  exit
end
theTheme = theThemes.Get(0)
if(theTheme.Is(FTheme).Not) then
  exit
end
theVTab = theTheme.GetFTab
if ((theVTab.GetShapeClass.GetClassName = "PolyLine") or
    (theVTab.GetShapeClass.GetClassName = "PolyLineM") or
    (theVTab.GetShapeClass.GetClassName = "PolyLineZ")) then
else
  exit
end

' Sicherheitshalber rückfragen

if(MsgBox.YesNo("Achtung!" + NL +

```

```

"Die Orientierung aller selektierten Polylinien" + nl + "im
Thema"++theTheme.AsString++
"wird geändert."+ nl +
"Sind Sie sicher ?", "SIS", false) = false) then
exit
end

theShpFld = theVTab.FindField ("Shape")

if (theVTab.GetNumSelRecords <> 0) then
  if(theVTab.CanEdit.Not) then
    MsgBox.Error("VTab ist nicht editierbar!", "")
    theTheme.ClearSelection
    exit
  end
  theVTab.SetEditable (true)
  for each rec in theVTab.GetSelection
    pl = theVTab.ReturnValue (theShpFld, rec)
    theVTab.SetValue (theShpFld, rec, pl.Flip)
  end
  theVTab.SetEditable (false)
  theTheme.ClearSelection
end

av.GetProject.SetModified(true)
av.ProcessAllInvals

'-----Scriptende-----

' Name: SIS.GetUserTheme
'
' Zweck: Ermöglicht die Auswahl eines Themas durch den Benutzer
'
'
'
' Thema: SIS, Utility
'
' Voraussetzung: Das aktive Dokument muss ein View sein.
'
'
' Self: (0) Shapetyp als Text
'         mögliche Werte:
'         Point, PolyLine, Polygon, oder MultiPoint
'         (1) Text der Auswahlbox
'
' Rückgabe: (0) Theme oder nil bei Fehler oder Abbruch
'
' Autor: ArcView GIS/Avenue Programmer's Reference
'
t="SIS.GetUserTheme"

' Shapetyp für Auswahl übernehmen

theShapeType = self.get(0)
theBoxTitle = self.Get(1)

if(theShapeType = nil) then
  MsgBox.Error("Übergabeparameter Shapetype fehlt!",t)
  return({nil})
end

' default-Titel

```

```

if(theBoxTitle = nil) then
  theBoxTitle = "Thema auswählen:"
end
theView = av.GetActiveDoc

tList = theView.GetThemes

' keine Themen im aktuellen View

if(tList.Count = 0) then
  MsgBox.Error("Keine Themen im aktuellen View!",t)
  return({nil})
end

' Themen auf gewünschten Typ filtern

fTList = {}

for each theTheme in tList
  if(theTheme.Is(FTheme)) then
    theFtab = theTheme.GetFtab
    if(theFtab.GetShapeClass.GetClassName = theShapeType) then
      fTList.Add(theTheme)
    end
  end
end

' Abbruch wenn Themen nicht vorhanden

if(fTList.Count = 0) then
  MsgBox.Error("Keine Themen vom Typ:"++ theShapeType ++ "im aktuellen
  View!",t)
  return({nil})
end

theChoose = MsgBox.List(fTList,theBoxTitle,"SIS")
if(theChoose = nil) then
  return({nil})
else
  return({theChoose})
end

'-----Scriptende-----

' Name: SIS.IsLine.Update
'
' Zweck: Überprüft ob es sich beim ausgewählten Thema um ein
'        dynamisch erzeugtes Linienthema handelt. Wird zur
'        Aktivierung des Toolbuttons verwendet
'
' Thema: SIS, Dynamische Sementierung
'
' Voraussetzung: Das aktive Dokument muss ein View sein. Das
'                Script muss als Updatefunktion verwendet werden.
'
' Self: Update
'
' Rückgabe:
'
' Autor: Horst Baumgartner 2000/02/13
'

theView = av.GetActiveDoc

```

```

if(theView.GetActiveThemes.Count = 1) then
    theTheme= theView.GetActiveThemes.Get(0)
    theSrcName = theTheme.GetSrcName

    if(theSrcName.Is(DynName)) then
        self.SetEnabled(theSrcName.IsLinear)
    else
        self.SetEnabled(false)
    end
end
else
    self.SetEnabled(false)
end

'-----Scriptende-----

' Name:   SIS.IsPoint.Update
'
' Zweck:  Überprüft ob es sich beim ausgewählten Thema um ein
'         dynamisch erzeugtes Punktthema handelt. Wird zur
'         Aktivierung des Toolbuttons verwendet
'
' Thema:  SIS, Dynamische Sementierung
'
' Voraussetzung: Das aktive Dokument muss ein View sein. Muss
'               als Updatefunktion verwendet werden.
'
' Self:   Update
'
' Rückgabe:
'
' Autor:  Horst Baumgartner 2000/02/13
'

theView = av.GetActiveDoc
if(theView.GetActiveThemes.Count = 1) then
    theTheme= theView.GetActiveThemes.Get(0)
    theSrcName = theTheme.GetSrcName

    if(theSrcName.Is(DynName)) then
        self.SetEnabled(theSrcName.IsPoint)
    else
        self.SetEnabled(false)
    end
end
else
    self.SetEnabled(false)
end

'-----Scriptende-----

'SIS.Kt.Lbt_Abbruch.Click
'Bricht den Dialog Kantendaten ab
'und gibt den Wert 0 zurück

self.GetDialog.SetModalResult(0)
self.GetDialog.Close

'-----Scriptende-----

'SIS.Kt.lbt_OK.Click
'Schließt die Dialogbox und gibt den Wert 1
'zurück

```

```

self.GetDialog.SetModalResult(1)
self.GetDialog.Close

'-----Scriptende-----

'SIS.Kt.lbx_StrVerz.Select
'Teil der ListBox in SisKt zur Ermittlung
'des statistischen Straßencodes

StrNr = self.GetSelection.Get(0)
txt_StrNr = self.GetDialog.FindByName("txt_Select")
txt_StrNr.SetText(StrNr.AsString)

'-----Scriptende-----

'SIS.Kt.Open
'Öffnet eine Dialogbox zur Kategorisierung und Bewertung
'einer Straßenkante
'
'Als Parameter ist eine Liste mit folgenden Daten zu übergeben:
'
'Number ID
'Number Straßencode
'Number Length
'Number Gattung
'Number Kategorie
'String Regel
'Number Geschwindigkeit
'String Alternativbezeichnung
'String Verordnung
'String User (derzeit nicht verwendet)
'Datum Date (derzeit nicht verwendet)
'
'Folgende Parameter werden in einer Liste zurückgegeben:
'Nil bei Abbruch, die anderen Werte sind nicht editierbar
'
'Number Straßencode
'Number Gattung
'Number Kategorie
'String Regel
'Number Geschwindigkeit
'String Alternativbezeichnung
'String Verordnung

'hier steht der Name der Tabelle mit dem Straßenverzeichnis
'-----
SVerzName = "strverz.dbf"
'-----

'Parameter des selektierten Straßenabschnitts

theDialog = av.FindDialog("SISKt")
theStrnum = theDialog.FindByName("txt_Select")
theStrnum.SetText(self.Get(1).AsString)
theLen = theDialog.FindByName("txt_Len")
theLen.SetText(self.Get(2).AsString)
theID = theDialog.FindByName("txt_ID")
theID.SetText(self.Get(0).AsString)
theRegel = theDialog.FindByName("cbx_Regel")
theRegel.DefineFromList({"", "FT", "TF", "N"})
theRegel.FindByValue(self.Get(5))
theRegel.SelectCurrent
theSpeed = theDialog.FindByName("txt_Speed")
theSpeed.SetText(self.Get(6).AsString)
theAltBez = theDialog.FindByName("txt_AltBez")

```

```

theAltBez.SetText(self.Get(7).AsString)
theVero = theDialog.FindByName("txl_Vero")
theVero.SetText(self.Get(8))
theSaveBut = theDialog.FindByName("lbt_OK")

' Buttons als Liste definieren um den Wert zu setzen wenn
' vorhanden

GatBut = {"rad_BS", "rad_LS", "rad_GS", "rad_Ort", "rad_Privat"}

if(self.Get(3) > 0) then
    theGat = theDialog.FindByName(GatBut.Get(self.Get(3)-1))
    theGat.Select
end

KatBut = {"rad_wieGat", "rad_Rad", "rad_FG", "rad_Wand", "rad_Guew"}

if(self.Get(4) > 0) then
    theKat = theDialog.FindByName(KatBut.Get(self.Get(4)-1))
    theKat.Select
end

'Straßenverzeichnis holen und in Listbox einfügen
aListBox = theDialog.FindByName("lbox_StrVerz")
aSVerzDoc = av.GetProject.FindDoc(SVerzName)
if(aSVerzDoc <> nil) then
    aVTab = aSVerzDoc.GetVTab
    myFields = aVTab.GetFields
    aListBox.DefineFromVTab(aVTab, myFields, false)
    aListBox.SortAscending(true)
    aListBox.FindByValue(self.Get(1).AsString)
    aListBox.SelectCurrent(true)
    aListBox.ShowCurrent
    theStrNum.SetReadOnly(true)
else
    theStrNum.SetReadOnly(false)
end

'den Speichern Button nur aktivieren, wenn das Thema editierbar ist

theView = av.GetActiveDoc
theTheme = theView.GetEditableTheme

if (theTheme <> nil) then
    theSaveBut.SetEnabled(true)
else
    theSaveBut.SetEnabled(false)
end

ret = theDialog.Open

'Auswerten der RadioButtons ???

if(ret = 1) then
    rbGat = theDialog.FindByName("cpa_Gat").GetSelected.AsString
    selGat = GatBut.FindByValue(rbGat) + 1
    rbKat = theDialog.FindByName("cpa_Kat").GetSelected.AsString
    selKat = KatBut.FindByValue(rbKat) + 1
    selRegel = theRegel.GetSelection
    return({theStrnum.GetText.AsNumber, selGat, selKat, selRegel,
    theSpeed.GetText.AsNumber, theAltBez.GetText, theVero.GetText})
else
    return(nil)
end

'-----Scriptende-----

```

```

' Name:   SIS.Kt.Recalc
'
' Zweck: Ergänzen von Länge und ID nach dem Editieren
'
'
' Thema:  SIS, Tabellenwerte
'
' Voraussetzung: Das aktive Dokument muss ein View sein. Thema
'                Die Felder ID, und Length müssen
'                in der FTab vorhanden sein.
' Self:
'
' Rückgabe: keine
'
' Autor:  Horst Baumgartner 2001/11/14'

isOK = MsgBox.LongYesNo("Achtung: Diese Funktion kann Werte in den
Feldern:"+ nl +
"ID, Length, Komb und Seconds verändern!" + nl + "Wollen Sie fortsetzen?",
                      "SIS-Recalc",true)

if(isOK = false)
  then exit
end

theView = av.GetActiveDoc
theCount = theView.GetActiveThemes.Count
If (theCount = 0) then
  msgbox.error("SisKt.Recalc:Kein Thema aktiv im aktuellen
  View.,"SIS-Error")
  exit
End

theTheme = theView.GetActiveThemes.Get(0)
theFTab = theTheme.GetFTab

'nur wenn das aktive Thema auch vom Typ Polylinie ist ...

theShapeField = theFTab.FindField("Shape")
if (theShapeField.GetType <> #FIELD_SHAPELINE) then
  MsgBox.Info("Das Thema muss vom Typ Polylinie sein.,"SIS")
  exit
end

theFTab.StartEditingWithRecovery

'Suchen nach den erforderlichen Feldern in der FTab und Erzeugen der Felder
falls
'diese noch nicht vorhanden sind

theIDField = theFTab.FindField("Id")
if(theIDField = nil) then
  theIDField = Field.Make("Id",#FIELD_DOUBLE,16,0)
  theFTab.AddFields({theIDField})
end

theLenField = theFTab.FindField("Length")
if(theLenField = nil) then
  theLenField = Field.Make("Length",#FIELD_DOUBLE,16,2)
  theFTab.AddFields({theLenField})
end

theGatField = theFTab.FindField("Gattung")
theKatField = theFTab.FindField("Kat")
theKombField = theFTab.FindField("Komb")

```

```

theSecField = theFTab.FindField("Seconds")
theSpeedField = theFTab.FindField("Speed")

' Längen neu rechnen und IDs ergänzen wenn erforderlich
' zuerst die nächste freie ID für die Kanten ermitteln

theNextID = av.Run("SIS.TabIndex",{theFTab,theIdField})

for each rec in theFTab
  theLen = theFTab.ReturnValue(theShapeField,rec).ReturnLength
  theID = theFTab.ReturnValue(theIDField,rec)
  theFTab.SetValue(theLenField,rec,theLen)

  ' Es kann sich ja um den ersten Datensatz handeln !
  ' Nullwert beim ersten Datensatz ist möglich

  if((rec > 0) and (theID = 0)) then
    theFTab.SetValue(theIdField,rec,theNextID)
    theNextID = theNextID + 1
  end

  ' Falls die notwendigen Felder existieren wird das Attribut berechnet

  if((theGatField <> nil) and (theKatField <> nil)) then
    if(theKombField <> nil) then
      theGatVal = theFTab.ReturnValue(theGatField, rec)
      theKatVal = theFTab.ReturnValue(theKatField, rec)
      theFTab.SetValue(theKombField, rec , (theGatVal.AsString + "." +
        theKatVal.AsString))
    end
  end

  if((theSpeedField <> nil) and (theSecField <> nil)) then
    theSpdVal = theFTab.ReturnValue(theSpeedField, rec)
    if(theSpdVal > 0) then
      theFTab.SetValue(theSecField, rec , ((theLen * 3.6) / theSpdVal))
    else
      theFTab.SetValue(theSecField, rec , 0)
    end
  end
end

theFTab.StopEditingWithRecovery(true)

'-----Scriptende-----

' Name:   SIS.Kt.Set
'
' Zweck: Auswahl einer Straßenkante und Informieren/Editieren über
'         Dialogbox SisKt
'
'
' Thema:  SIS, Tabellenwerte
'
' Voraussetzung: Das aktive Dokument muss ein View sein. Thema
'                 Die Felder ID, StrNum, Length, Gattung, Kat, Oneway,
Speed,
'                 AltBez, Vero, User und Date müssen
'                 in der FTab vorhanden sein.
' Self:
'
' Rückgabe: keine
'
' Autor:  Horst Baumgartner 2000/02/19
'

```

```

theView = av.GetActiveDoc
theCount = theView.GetActiveThemes.Count
If (theCount = 0) then
  msgbox.error("SisKt.Set:Kein Thema aktiv im aktuellen View.,"SIS-Error")
  exit
End

theDisplay = theView.GetDisplay

theTheme = theView.GetActiveThemes.Get(0)
theFtab = theTheme.GetFtab
theClassName = theFtab.GetShapeClass.GetClassName.AsString

'nur wenn das aktive Thema auch vom Typ Polylinie ist ...

If (theClassName <> "PolyLine") then
  Exit
end

'Suchen nach den erforderlichen Feldern in der FTab
'Abbruch beim Fehlen eines der Felder

FieldNames =
{"Id", "Strnum", "Length", "Gattung", "Kat", "Oneway", "Speed", "AltBez", "Vero", "U
ser", "Date"}
Fields = List.Make

for each fnam in FieldNames
  theField = theFtab.FindField(fnam)
  if(theField <> nil) then
    Fields.Add(theField)
  else
    MsgBox.Error("SisKt.Set:Feld"++ fnam ++"nicht vorhanden!","SIS-Error")
    Exit
  end
end

theSPoint = theDisplay.ReturnUserPoint
theFtab.SelectByPoint(theSPoint, 5, #VTAB_SELTYPE_NEW)

'soll gleich sichtbar werden

av.ProcessAllInvals

'Selektion auswerten

theBitmap = theFtab.GetSelection

if(theBitmap.Count = 0) then
  Exit
end

'Nur, wenn auch etwas ausgewählt wurde

rec = theBitMap.getNextSet(-1)

' Polylineshape abfragen
Param = List.Make

for each col in Fields
  Param.Add(theFtab.ReturnValue(col,rec))
end

```

```

res = av.Run("SIS.Kt.Open",Param)

'Dialog wurde abgebrochen, Werte bleiben unverändert

if(res = nil) then
    exit
end

'Tabellenfelder aktualisieren
'wenn nicht schon zum Editieren geöffnet dann jetzt

if(theFtab.IsBeingEditedWithRecovery) then
    theFtabisEdit = true
else
    theFtab.StartEditingWithRecovery
    theFtabisEdit = false
end

'Die Felder stehen noch in der Fields-Liste

'Straßencode
theFtab.SetValue(Fields.Get(1),rec,res.Get(0))
'Gattung
theFtab.SetValue(Fields.Get(3),rec,res.Get(1))
'Kategorie
theFtab.SetValue(Fields.Get(4),rec,res.Get(2))
'Regel
theFtab.SetValue(Fields.Get(5),rec,res.Get(3))
'Geschwindigkeit
theFtab.SetValue(Fields.Get(6),rec,res.Get(4))
'Alternativbezeichnung
theFtab.SetValue(Fields.Get(7),rec,res.Get(5))
'Verordnung
theFtab.SetValue(Fields.Get(8),rec,res.Get(6))
'Benutzerkennung festhalten
theFtab.SetValue(Fields.Get(9), rec, _theUser)

theDate = Date.Now
theDate.SetFormat("yyyy-mm-dd")
'Datum festhalten
theFtab.SetValue(Fields.Get(10), rec, theDate)

' Für den Fall, dass das Editieren vorher noch nicht möglich war

if(theFtabisEdit = false) then
    theFtab.StopEditingWithRecovery(True)
end

av.GetProject.SetModified(true)

'-----Scriptende-----

' Name:  Sis.KtUnion
'
' Title:  Kanten zusammenfügen
'
' Topics:
'
' Description:  Fügt zwei oder mehrere Kanten zu einer
'               Kante zusammen
'

```

```

' Requires: View mit mindestens einem editierbaren Thema
'
' Self:
'
' Returns:

theView = av.GetActiveDoc
theTheme = theView.GetEditableTheme

theFtab = theTheme.GetFtab
theUserField = theFtab.FindField("User")
theDateField = theFtab.FindField("Date")
theShapeField = theFtab.FindField("Shape")
theLenField = theFtab.FindField("Length")

' Kanten zuerst auf ungültig setzen
av.run("SIS.StrInvalid",nil)

theMergedPoly = PolyLine.Make({{}})

theTheme.GetFtab.BeginTransaction

for each rec in theFtab.GetSelection
  theShape = theFtab.ReturnValue(theShapeField,rec)
  theMergedPoly = theMergedPoly.ReturnMerged(theShape.AsPolyline)
end

rec = theFtab.AddRecord

theFtab.SetValue(theShapeField,rec,theMergedPoly)

if(theUserField <> nil) then
  theFtab.SetValue(theUserField,rec,"baum")
end

if(theDateField <> nil) then
  theFtab.SetValue(theDateField,rec,Date.Now)
end

if(theLenField <> nil) then
  theFtab.SetValue(theLenField,rec,theMergedPoly.ReturnLength)
end

theTheme.GetFtab.EndTransaction
theTheme.Invalidate(TRUE)

'----Scriptende----

' Name:  Sis.KtUnionUpdate
'
' Title:  Updatefunktion für SIS.KtUnion
'
' Topics:
'
' Description:  Prüft ob das Thema editierbar ist und mehr als ein
                Objekt ausgewählt
'
'
' Requires:  View mit mindestens einem editierbaren Thema
'
' Self:
'
' Returns:

```

```

theView = av.GetActiveDoc
theTheme = theView.GetEditableTheme
enabled = FALSE

if (theTheme <> nil) then
  fCount = theTheme.GetFTab.GetSelection.Count
  enabled = (fcount > 1)
end

SELF.SetEnabled(enabled)

'-----Scriptende-----

' Name:   SIS.ListSub
'
' Zweck: Restmenge von Listenelementen, wobei
'        doppelte Elemente aus der ersten Liste erhalten
'        bleiben
'        z.B.: { 1, 2, 2, 3 } - {1, 2, 3} = {2}
'        In Original Avenue = {}
'
'
'
' Thema: SIS, Dynamische Sementierung, Listen
'
' Voraussetzung:
'
' Self: Param 1: theFList
'        Param 2: theSList
'
' Rückgabe: Liste
'
' Autor: Horst Baumgartner 2000/02/19
'

theFList = self.Get(0).DeepClone
theSList = self.Get(1)

for each elem in theSList
  ix = theFList.FindByValue(elem)
  if(ix <> nil) then
    theFList.Remove(ix)
  end
end

theFList.RemoveDuplicates

return(theFList)

'-----Scriptende-----

' Name:   SIS.LnDefine.Read
'
' Zweck: Liest die Einstellungen aus der Datei: $HOME/SIS.def
'        aus und gibt eine Liste mit den Parametern zurück
'
'
'
' Thema: SIS, Dynamische Sementierung
'
' Voraussetzung:
'
' Self:
'
'

```

```

' Rückgabe: Parameter 0: Name des Bezugspunktthemas
'           1: Name des Bezugsroutenthemas
'           2: Name der Dyn.Segmenttabelle
'           3: Name der Dyn.Routentabelle
'           oder nil im Fehlerfall
'
' Autor: Horst Baumgartner 2000/02/22
'

thePMList = List.Make

theSisFile = LineFile.Make("$Home/SIS.def".AsFileName,#FILE_PERM_READ)

if (theSisFile = nil) then
  MsgBox.Error("Definitionsdatei kann nicht gelesen werden!","SIS-Route")
  exit
end

while(theSisFile.IsAtEnd.Not)
  thePMList.Add(theSisFile.ReadElt)
end

theSisFile.Close

if(thePMList.Count = 4) then
  return(thePMList)
else
  return(nil)
end

'-----Scriptende-----

' Name:   SIS.LnDefine.Write
'
' Zweck: Definiert jene Themen und Tabellen die für
'         die Segmentierung von Linien verwendet werden sollen
'         und sichert die Einstellungen in der Datei: $HOME/SIS.def
'
'
' Thema: SIS, Dynamische Segmentierung
'
' Voraussetzung: Avenuescripts SIS.ViewDP.List,SIS.Table.List
'                und SIS.Routthm.List
'
' Self:
'
' Rückgabe:
'
' Autor: Horst Baumgartner 2000/02/22
'

'Parameter in einer Liste Speichern

thePMList = List.Make

thePMList.Add(MsgBox.ListAsString(av.Run("SIS.ViewDP.List",0),"Bezugs-Punkt
thema auswählen:","SIS-Linienthemen"))
if(thePMList.get(0) = nil) then exit end
' Das Routenthema kann ev. auch aus dem Punktthema bestimmt werden
'
MsgBox.Info(av.getActiveDoc.FindTheme((thePMList.Get(0))).GetSrcName.GetRou
teSys.GetSrcName.AsString,"")
' Durchsuchen des Views wäre notwendig... nächste Version
thePMList.Add(MsgBox.ListAsString(av.Run("SIS.PLineM.List",0),"Bezugs-Route
nthema auswählen:","SIS-Linienthemen"))
if(thePMList.get(1) = nil) then exit end

```

```

thePMList.Add(MsgBox.ListAsString(av.Run("SIS.Table.List",0),"Dyn.Segmentab
elle auswählen:", "SIS-Linienthemen"))
if(thePMList.get(2) = nil) then exit end
thePMList.Add(MsgBox.ListAsString(av.Run("SIS.Table.List",0),"Dyn.Routentab
elle auswählen:", "SIS-Linienthemen"))
if(thePMList.get(3) = nil) then exit end

theSisFile = LineFile.Make("$Home/SIS.def".AsFileName,#FILE_PERM_WRITE)

if (theSisFile = nil) then
  MsgBox.Error("Definitionsdatei kann nicht geschrieben
werden!", "SIS-Route")
  exit
end

for each str in thePMList
  theSisFile.WriteElt(str)
end

theSisFile.Close

'----Scriptende-----

' Name: SIS.MeasurePolyline
'
' Zweck: Erzeugt ein neues PolylineM-Thema aus einem Polylinethema
'        um eine dynamische Segmentierung zu ermöglichen
'
'
' Thema: SIS, Dynamische Sementierung
'
' Voraussetzung: Das aktive Dokument muss ein View sein. Das Polylinethema
'                muß aktives Thema sein.
'
' Self:
'
' Rückgabe:
'
' Autor: ArcView-Online-Help
'
' This is a simple script that will convert a PolyLine theme to a PolyLineM
' theme. The measures will not be explicitly set and will, therefore, all be
' Null Numbers. You can use the field calculator along with requests such as
' SetMASLength to add measure values.
'
' You can add this script to a button in the View as a CLICK script. Only
' the first active theme will be processed. The active theme must be
' PolyLine.
'
' Any selected set will be ignored by this script.
'
' Get the active theme ...

theView = av.GetActiveDoc
theCount = theView.GetActiveThemes.Count

If (theCount = 0) then
  msgbox.info("There is no active theme to process.", "No active theme...")
  exit
End

```

```

theTheme = theView.GetActiveThemes.Get(0)
theFtab = theTheme.GetFtab
theClassName = theFtab.GetShapeClass.GetClassName.AsString

'Only proceed with the active theme if it is PolyLine ...

If (theClassName = "PolyLine") then
  theFieldList = theFtab.GetFields.Clone

  theShapeField = theFtab.FindField("shape")

  'Create a new shapefile with fields ...

  theNewShapefile = Filedialog.Put("*.shp".AsFilename,"*.shp",
    "Enter new shapefile name for "+theTheme.GetName)
  If (theNewShapefile = nil) then
    exit
  End
  theNewFtab = Ftab.MakeNew(theNewShapefile,PolyLineM)

  ' StartEditingWithRecovery führt bei größeren Datenbeständen zu Abstürzen

  theNewFtab.SetEditable(true)

  'Populate the new shapefile with Fields ..

  theFieldList.RemoveObj(theShapeField) 'ignore the shape field

  theFieldCount = theFieldList.Count - 1
  theNewShapeField = theNewFtab.FindField("shape")
  theNewFieldList = theFieldList.DeepClone
  theNewFtab.AddFields(theNewFieldList)

  'Convert the PolyLines to PolyLineMs and add the measures ...

  For each rec in theFtab
    theShape = theFtab.ReturnValue(theShapeField,rec)
    theShapeM = theShape.AsPolyLineM

    ' Und Stationieren

    theShapeM.SetMAsLength

    'Get the attributes back ...

    theNewRecnum = theNewFtab.AddRecord
    theNewFtab.SetValue(theNewShapeField,theNewRecnum,theShapeM)

    For each fld in (0..theFieldCount)
      theVal = theFtab.ReturnValue(theFieldList.Get(fld),rec)
      theNewFtab.SetValue(theNewFieldList.Get(fld),theNewRecnum,theVal)
    End
  End 'For each rec...
Else 'If the current theme is not a PolyLine theme ...
  msgbox.info (theTheme.GetName+" is a "+theClassName.AsString+" theme, not
  a PolyLine theme.", theTheme.GetName+" is not a PolyLine theme")
  exit
End 'If (theClassName = "PolyLine") ...

theNewFtab.SetEditable(false)

'Add the new shapefile to the view if the user wants...

If (MsgBox.YesNo("Add shapefile as theme to the view?",
  "Convert to Shapefile",true)) then
  newFtheme = Ftheme.Make(theNewFtab)

```

```

    theView.Addtheme(newFtheme)
    newFtheme.Setvisible(true)
End

'-----Scriptende-----

' Name:   SIS.MkFTheme
'
' Zweck:  Thema neu erstellen und zu einem View hinzufügen
'
'
' Thema:  SIS, Werkzeuge
'
' Voraussetzung: Zu übergeben sind neben dem Titel der Dialogbox und
'                einem Default-Dateinamen eine Liste mit Elementen vom Typ
Field.
'                Das aktive Dokument muss vom Typ View sein.
'
'
' Self: Dialogboxtitel      0
'       Default-Dateiname  1
'       Field-Liste        2
'       Shape-Typ          3
'
' Rückgabe: keine
'
' Autor:  Horst Baumgartner 2001/11/21
'
'
if(self.Count <> 4) then
    exit
end

theView = av.GetActiveDoc

theTitle=self.Get(0)
theFnamStr = self.Get(1)
theFieldList = self.Get(2)
theShapeType = self.Get(3)

if(theFieldList.HasKindOf(Field).Not) then
    MsgBox.Error("Der Funktion SIS.MkFtheme wurde keine Feld-Liste
übergeben.", "SIS")
    exit
end

theShapeName = FileDialog.Put(theFnamStr.AsFilename, "*.shp", theTitle)

' bei Abbruch

if(theShapeName <> nil) then

    theFTab = FTab.MakeNew(theShapeName, theShapeType)
    if (theFTab.HasError) then
        if (theFTab.HasLockError) then
            MsgBox.Error("Unable to acquire Write Lock for file " +
theShapeName.GetBaseName, "")
        else
            MsgBox.Error("Unable to create " + theShapeName.GetBaseName, "")
        end
        return nil
    end
end
end

```

```

' Felder hinzufügen

theFtab.AddFields(theFieldList)
theFtab.SetEditable(False)

' Shapefile dem View hinzufügen

newFtheme = Ftheme.Make(theFtab)
theView.Addtheme(newFtheme)
newFtheme.SetActive(true)
newFtheme.Setvisible(true)
theView.SetEditableTheme(newFtheme)
av.GetProject.SetModified(true)
end

'-----Scriptende-----

' Name: SIS.MkRandomPoint
'
' Zweck: Setzt eine gewisse Anzahl von Zufallspunkten auf einem Linienthema
'        und zwar am Mittelpunkt zufällig ausgewählter Linien.
'
'
'
' Thema: SIS, Utility
'
' Voraussetzung:
'
' Self:
'
' Rückgabe: Es wird ein Punktthema erstellt
'
'
' Autor: Baumgartner
'
'

t= "SIS-Zufallspunkte"

' Linienthema auswählen

theKTheme = av.run("SIS.GetUserTheme",{ "Polyline", "Linienthema
auswählen"}).get(0)

if(theKTheme = nil) then
  MsgBox.Info("Script wurde vom Benutzer abgebrochen!",t)
  exit
end

theKFTab = theKTheme.getFtab

theKANz = theKFTab.GetNumRecords

' gewünschte Punktanzahl ermitteln

theAnzStr = MsgBox.Input("Kantenzahl="++ theKANz.AsString + nl +"Gewünschte
Punktanzahl eingeben:",t,"100")

if((theAnzStr = nil) or (theAnzStr.IsNumber.not)) then
  exit
else
  theAnz = theAnzStr.AsNumber
  if(theAnz > theKANz) then

```

```

    theAnz = theKANz
  end
end

theKShapeField = theKFtab.FindField("Shape")

' nur zum Testen

d = av.GetactiveDoc

av.ClearStatus

for each i in 1..theAnz
  theRandomNum = Number.MakeRandom(0,theKANz-1)
  theShape = theKFtab.ReturnValue(theKShapeField,theRandomNum)
  xShape = GraphicShape.Make(theShape.ReturnCenter)
  xShape.GetSymbol.SetSize(8)
  d.GetDisplay.BeginClip
  d.GetGraphics.Add(xShape)
  d.GetDisplay.EndClip
  av.ShowMsg(i.AsString+".Punkt")
end

av.ClearStatus
d.GetDisplay.Invalidate(true)

'-----Scriptende-----

' Name:  SIS.MkTable
'
' Zweck: Erstellen einer Tabelle und Anlegen von Feldern
'
'
'
' Thema: SIS, Werkzeuge
'
' Voraussetzung: Zu übergeben sind neben dem Titel der Dialogbox und
'                einem Default-Dateinamen eine Liste mit Elementen vom Typ
Field.
'                Abschließend wird die Tabelle zur Bearbeitung geöffnet,
wenn
'                openForEdit den Wert true enthält.
'
'
' Self: Dialogboxtitel      0
'       Default-Dateiname  1
'       Field-Liste        2
'       zum Editieren öffnen 3
'
' Rückgabe: keine
'
' Autor: Horst Baumgartner 2001/11/21
'
'

if(self.Count <> 4) then
  exit
end

theTitle=self.Get(0)
theFnamStr = self.Get(1)
theFieldList = self.Get(2)
openForEdit = self.Get(3)

if(theFieldList.HasKindOf(Field).Not) then

```

```

    MsgBox.Error("Der Funktion SIS.MkTable wurde keine Feld-Liste
    übergeben.", "SIS")
    exit
end

theFName = FileDialog.Put(theFnamStr.AsFilename, "*.dbf", theTitle)

' bei Abbruch

if(theFName = nil) then
    exit
end

theVTab = VTab.MakeNew(theFName, dbase)
theTable = Table.Make(theVTab)

theTable.SetName(theFName.GetBaseName)

theVtab.AddFields(theFieldList)

' Tabelle anzeigen

if(openForEdit = true) then
    theVTab.StartEditingWithRecovery
    theTable.GetWin.Open
else
    theVTab.SetEditable(FALSE)
end

'-----Scriptende-----

' Name:   SIS.MkTurnTable
'
' Title:  Erstellt aus einem Punktthema und einem dazugehörigen Linienthema
'         mit Network Index Directory eine Tabelle mit Abbiegeregeln
'
' Topics: SIS, Netzwerk, Network-Analyst
'
' Description: Aus einem Punktthema mit Abbiegebeziehungen und einem
'             dazugehörigen Linienthema mit Network Indey Directory wird
'             eine Tabelle (turntable) erzeugt, die den Anforderungen des
'             Network-Analyst entspricht
'
'
' Requires: View mit einem Punkt- und einem Linienthema
'
' Self:
'
' Returns:

theTTheme = av.Run("SIS.GetUserTheme", {"Point", "Abbiegethema
auswählen"}).Get(0)
if(theTTheme = nil) then
    exit
end

theTTftab = theTTheme.GetFtab

for each fld in {"F_EDGE", "T_EDGE", "SECONDS"}
    if(theTTftab.FindField(fld) = nil) then
        MsgBox.Error("Feld:[" + fld + "] fehlt in der Attributtabelle.", "SIS")
        exit
    end
end
end

```

```

theShapes = theTTFTab.FindField("Shape")
FedgeFlds = theTTFTab.FindField("F_EDGE")
TedgeFlds = theTTFTab.FindField("T_EDGE")
SecondsFlds = theTTFTab.FindField("SECONDS")

theLTheme = av.Run("SIS.GetUserTheme",{ "Polyline", "Linienthema
auswählen" }).Get(0)
if(theLTheme = nil) then
    exit
end

theLFTab = theLTheme.GetFtab

theIDField = theLFTab.FindField("ID")
theShapeFld = theLFTab.FindField("Shape")

if(theIDField = nil) then
    MsgBox.Error("Feld:[ID] fehlt in der Attributtabelle.", "SIS")
    exit
end

' Suche nach der Datei nodes.dbf, Erzeugen eines VTab-Objektes, und
Definition
' der Felder

theNwIndexDir =
theLTheme.GetSrcName.GetFilename.AsString.Substitute("shp", "nws")
theNodeFile = Filename.Merge(theNwIndexDir, "nodes.dbf")

if(VTab.CanMake(theNodeFile)) then
    theNodeVTab = VTab.Make(theNodeFile, false, false)
    aFjunction = theNodeVTab.FindField("Fjunction")
    aTjunction = theNodeVTab.FindField("Tjunction")
else
    MsgBox.Error("Tabelle"++theNodeFile.AsString++"kann nicht geöffnet
werden.", "SIS")
    exit
end

' Erzeugen einer Network-Analyst kompatiblen Tabelle mit den
' erforderlichen Feldern

theNwTTFFName = FileDialog.Put("ttbl.dbf".AsFilename, "*.*", "Neue
Abbiegetabelle")

' Abbruch wenn die Tabelle nicht definiert sein sollte

if(theNwTTFFName = nil) then
    exit
end

theNwTTVTab = VTab.MakeNew(theNwTTFFName, dbase)
theNwTTTable = Table.Make(theNwTTVTab)

theNwTTTable.SetName(theNwTTFFName.GetBaseName)

' Felder für die Turntable
JunctionFld = Field.Make("JUNCTION", #FIELD_DOUBLE, 12, 0)
FedgeFld = Field.Make("F_EDGE", #FIELD_DOUBLE, 12, 0)
ToedgeFld = Field.Make("T_EDGE", #FIELD_DOUBLE, 12, 0)
SecondsFld = Field.Make("SECONDS", #FIELD_DOUBLE, 12, 0)

theNwTTVTab.AddFields({ JunctionFld, FedgeFld, ToedgeFld, SecondsFld })

```

```

' Dictionary für die Punktsuche anlegen

pDict = Dictionary.Make(4)

' Abbiegethema bis zum letzten Datensatz lesen

theBitmap = theLFTab.GetSelection

for each rec in theTTFTab

    ' Suche die Von-Kante im Linienthema

    theFEdge = theTTFTab.ReturnValue(FedgeFlds, rec)
    expr = "([Id] = " + theFEdge.AsString + ")"
    theLFTab.Query(expr, theBitmap, #VTAB_SELTYPE_NEW)

    ' frec ist die Position der Von-Kante im Linienthema

    frec = theBitmap.GetNextSet(-1)
    if(frec = -1) then
        MsgBox.Error("Kante mit der ID:++ theFEdge.AsString ++"wurde nicht
            gefunden.", "SIS")
        continue
    end

    ' Suche die Nach-Kante im Linienthema

    theTEdge = theTTFTab.ReturnValue(TedgeFlds, rec)
    expr = "([Id] = " + theTEdge.AsString + ")"
    theLFTab.Query(expr, theBitmap, #VTAB_SELTYPE_NEW)

    ' frec ist die Position der Von-Kante im Linienthema

    trec = theBitmap.GetNextSet(-1)
    if(trec = -1) then
        MsgBox.Error("Kante mit der ID:++ theFEdge.AsString ++"wurde nicht
            gefunden.", "SIS")
        continue
    end

    ' Start- und Endpunkte der beiden Kanten bestimmen

    theFShape = theLFTab.ReturnValue(theShapeFld, frec)
    theTShape = theLFTab.ReturnValue(theShapeFld, trec)

    spf = theFShape.Along(0)
    epf = theFShape.Along(100)

    spt = theTShape.Along(0)
    ept = theTShape.Along(100)

    ' Punktnummern des Network-Analyst holen

    spfn = theNodeVTab.ReturnValue(aFjunction, frec)
    epfn = theNodeVTab.ReturnValue(aTjunction, frec)

    sptn = theNodeVTab.ReturnValue(aFjunction, trec)
    eptn = theNodeVTab.ReturnValue(aTjunction, trec)

pDict.Empty

'Punkte in String umwandeln
'zwar nicht sehr elegant, aber notwendig um sie wieder zu finden

pDict.Add(spf.AsString, spfn)

```

```

pDict.Add(epf.AsString, epfn)
pDict.Add(spt.AsString, sptn)
pDict.Add(ept.AsString, eptn)

' Punktnummer des Knotens aufgrund der Lage feststellen

theNodeNumber = pDict.Get(theTTFTab.ReturnValue(theShapes, rec).AsString)

if(theNodeNumber = nil) then
  MsgBox.Error("Fehler bei Kreuzung von [ID]:"++theFEdge.AsString++
  "nach [ID]:"++theTEdge.AsString,"SIS")
  continue
else
  ' Datensatz an neue Turntable anfügen und Werte aus dem Punktthema
  übernehmen
  nrec = theNwTTVTab.AddRecord
  theNwTTVTab.SetValue(FEdgeFld,nrec, frec+1)
  theNwTTVTab.SetValue(ToEdgeFld,nrec, trec+1)
  theNwTTVTab.SetValue(SecondsFld, nrec,
  theTTFTab.ReturnValue(SecondsFlds, rec))
  theNwTTVTab.SetValue(JunctionFld, nrec, theNodeNumber)
end
end

theBitmap.ClearAll
theNwTTVtab.SetEditable(FALSE)

'-----Scriptende-----

'Name: SIS.MPtoP
'
'      Purpose: Converts the active Multipoint-Feature theme in a View
into a Point-Feature theme.
'
'      Converts only selected records, if there are any. Else all
records will be converted.
'
'      The assigned values of the Multipoint-shapes can be selected and
assigned to each single Point
'
'      Author: Heiko Apel, Technical University of Braunschweig, Germany
'      email: h.apel@tu-bs.de
'
'
'
'
theView = av.GetActiveDoc
theTheme = theView.GetActiveThemes.Get(0)

if (theTheme.Is(FTheme)) then

theFTab = theTheme.GetFTab

def = av.GetProject.MakeFileName("theme", "shp")
def = FileDialog.Put(def, "*.shp", "Convert " + theTheme.getName + " and
save as...")

if (def <> NIL) then

theNewFTab = Ftab.MakeNew(def, POINT)
theNewFTab.SetEditable(TRUE)

```

```

FieldList = MsgBox.MultiList(theFTab.GetFields, "Select the fields which
should be copied into the new Theme/Table!", "Field Selection")
FieldList = theFTab.GetFields-{theFTab.FindField("Shape")}

if (theNewFTab.CanAddFields = TRUE) then
  theNewFTab.AddFields(FieldList.deepclone)
end

if (theNewFTab.CanAddRecord = TRUE) then

  if (theFTab.GetSelection.Count >0) then
    for each i in theFTab.GetSelection
      dict = Dictionary.Make(10)
      for each t in theFTab.GetFields
        if (t.IsTypeShape = TRUE) then
          vallist = theFTab.ReturnValue(t, i).AsList
        else
          val = theFTab.ReturnValue(theFTab.FindField(t.AsString), i)
          Dict.Add(t, val)
        end
      end
      if (theNewFTab.HasError = FALSE) then
        for each z in vallist
          newrec = theNewFTab.AddRecord
          theNewFTab.SetValue(theNewFTAB.FindField("Shape"), newrec, z)
          for each u in Dict.ReturnKeys
            theNewFTab.SetValue(theNewFTab.FindField(u.AsString),
            newrec, Dict.Get(u))
          end
        end
      else
        msgbox.Error("FTAB has ERROR", "")
      end
    end
  else
    for each i in theFTab
      dict = Dictionary.Make(10)
      for each t in theFTab.GetFields
        if (t.IsTypeShape = TRUE) then
          vallist = theFTab.ReturnValue(t, i).AsList
        else
          val = theFTab.ReturnValue(theFTab.FindField(t.AsString), i)
          Dict.Add(t, val)
        end
      end
      if (theNewFTab.HasError = FALSE) then
        for each z in vallist
          newrec = theNewFTab.AddRecord
          theNewFTab.SetValue(theNewFTAB.FindField("Shape"), newrec, z)
          for each u in Dict.ReturnKeys
            theNewFTab.SetValue(theNewFTab.FindField(u.AsString),
            newrec, Dict.Get(u))
          end
        end
      else
        msgbox.Error("FTAB has ERROR", "")
      end
    end
  end
end

theNewFTab.SetEditable(false)

```

```

theNewFTheme=FTheme.Make(theNewFTab)
theView.AddTheme(theNewFTheme)
theNewFTheme.SetVisible(TRUE)

else
  msgbox.Error("Cannot add fields th new FTab!"+"\n"+"If problems continue,
  contact programmer <h.apel@tu-bs.de>.", "")
end

end

else msgbox.Error("Active Theme must be a Feature Theme!", "") end

'----Scriptende-----

' Name:  SIS.PLineM.List
'
' Zweck: Ermittelt alle Themen vom Typ PolylineM
'        im aktuellen View und gibt eine entsprechende
'        Liste zurück
'
'
' Thema: SIS, Dynamische Sementierung, Listen
'
' Voraussetzung:
'
' Self:
'
'
' Rückgabe: Liste
'
' Autor: Horst Baumgartner 2000/02/22
'

theView = av.GetActiveDoc
theThemes = theView.GetThemes

thePLineMList = List.Make
for each i in theThemes
  if(i.Is(FTheme)) then
    shapeType = i.GetFTab.GetShapeClass.GetClassName

    if (shapeType = "PolyLineM") then
      thePLineMList.Add(i.GetName)
    end
  end
end
return(thePLineMList)

'----Scriptende-----

' Name:  SIS.PtToLineTbl
'
' Zweck: Erstellt eine BP-Tabelle von einem Punktthema, wobei als
Referenzsystem das angegebene
'        Linienthema verwendet wird.
'
'
'
' Thema: SIS, Utility
'
' Voraussetzung: Das Linienthema muss das numerische Feld ID enthalten.
'                Das Punktthema sollte das String-Feld Info enthalten.
'                Das aktive Dokument muss ein View sein.
'                Sis.ReturnLSeqAzimuth wird aufgerufen um den Winkel für

```

```

die
'
'           Rotation des Einfügesymbols zu ermitteln
'
'
' Self:
'
' Rückgabe: Es wird eine Tabelle mit der Stationierung und dem Offfest
'           der Punkte erstellt.
'
' Autor: Baumgartner
'
'

t = "SIS.PtToLineTbl"

'

thePTheme = av.Run("SIS.GetUserTheme",{ "Point", "Punktthema
auswählen:"}).Get(0)
theLTheme = av.Run("SIS.GetUserTheme",{ "Polyline", "Linienthema
auswählen:"}).Get(0)

' Benutzerabbruch

if((thePTheme = nil) or (theLTheme = nil)) then
  exit
end

' Schüsselfeld aus dem Linienthema auswählen'
' l1 !!!

theLFTab = theLTheme.getFTab

l1 = MsgBox.List(theLFTab.GetFields,"Linien-Schlüsselfeld
auswählen:", "SIS")

if(l1 = nil) then
  MsgBox.Info("Das Linienthema muss ein numerisches ID-Feld enthalten!",t)
  exit
else
  If(l1.IsTypeNumber.Not) then
    MsgBox.Info("Das Linienthema muss ein numerisches ID-Feld
enthalten!",t)
    exit
  end
end

' Abbruch wegen Verwechslungsgefahr

if((thePTheme.GetFTab.FindField(l1.AsString) = nil).not) then
  MsgBox.Error("Das Punktthema darf kein Feld mit der Bezeichnung"++
l1.AsString ++"enthalten!",t)
  exit
end

' Tabellennamen festlegen

myFile =
FileDialog.Put("station.dbf".AsFileName,"*.dbf", "Stationierungstabelle")

' Benutzerabbruch

if(myFile = nil) then
  exit

```

```

end

theVTab = Vtab.MakeNew(myFile,dbase)
theTable = Table.Make(theVtab)
theTable.SetName(myFile.GetBaseName)

eFlst = List.Make

' Neue Felder der Exporttabelle erstellen

eFlst.Add(Field.Make("PID",#FIELD_LONG, 16, 0))
eFlst.Add(Field.Make("Str_id",#FIELD_LONG, 16, 0))
eFlst.Add(Field.Make("Stat",#FIELD_DOUBLE, 16, 2))
eFlst.Add(Field.Make("Offset",#FIELD_DOUBLE, 16, 2))
eFlst.Add(Field.Make("Rot",#FIELD_DOUBLE,10,4))

theVTab.AddFields(eFlst)

' Linienthema und Punktthema mit Join verbinden

' Punkt-Ftab holen

thePFTab = thePTheme.GetFTab

' Speichern der Felder

theFieldList = thePFTab.GetFields.Clone

thePShpFld = thePFTab.FindField("Shape")

' nur das Shapefeld aus der Liste entfernen

theFieldList.RemoveObj(thePShpFld)

' Felder des Punktthemas kopieren

theNewFieldList = theFieldList.DeepClone

theVTab.AddFields(theNewFieldList)

theLShpFld = theLFTab.FindField("Shape")

thePFTab.Join(thePShpFld,theLFTab,theLShpFld)

' ---

if(Not(theLFTab.IsFieldIndexed(11))) then
  theLFTab.CreateIndex(11)
end

theBitmap = theLFTab.GetSelection

theBitmap.ClearAll

' Straßenid ist jetzt beim Punktthema

theJID = thePFTab.FindField(11.AsString)
if(theJID = nil) then
  MsgBox.Error("Gejointes ID Feld wurde nicht gefunden!",t)
  exit
end

for each prec in thePFTab
  SuchSeg = thePFTab.ReturnValue(theJID, prec)

```

```

expr = "[" + l1.AsString + "]" = " + SuchSeg.AsString + "]"
theLftab.Query(expr, theBitMap, #VTAB_SELTYPE_NEW)
lrec = theBitMap.GetNextSet(-1)
if(lrec = -1) then
  break ' kein datensatz mehr vorhanden
end
PShp = thePftab.ReturnValue(thePShpFld, prec)
LShp = theLftab.ReturnValue(theLShpFld, lrec)
theLen = LShp.ReturnLength
perc = LShp.PointPosition(PShp.ReturnCenter)
offs = (LShp.QueryPointDistance(PShp,9999)) * (-1)
rec = theVTab.AddRecord
theVTab.SetValue(eFlst.Get(0), rec, rec)
theVTab.SetValue(eFlst.Get(1), rec, SuchSeg)
theVTab.SetValue(eFlst.Get(2), rec, ((theLen*perc)/100))
theVTab.SetValue(eFlst.Get(3), rec, offs)
theVTab.SetValue(eFlst.Get(4), rec,
av.Run("SIS.ReturnLSegAzimuth",{PShp.ReturnCenter,LShp.Along(perc)}))

for each fld in theNewFieldList
  theVTab.SetValue(fld, rec,
  thePftab.ReturnValue(thePftab.FindField(fld.AsString),prec))
end

end

theVTab.SetEditable(false)

theBitMap.ClearAll

'Join wieder entfernen

thePftab.UnJoinAll

'-----Scriptende-----

' Name: SIS.ReturnLSegAzimuth
'
' Title:
'
' Topics:
'
' Description: Calculates the angle created by two points.
'
' Requires:
'
' Self: See body of script.
'
' Results: Returns the angle (azimuth) formed by the two points as a
number from 0 - 360.
'
' Author: David F. Kimball

'-----
'the script must be passed a list containing either 2 points or 4 numbers:
'Sample script call: av.Run("SIS.ReturnLSegAzimuth",{1,1,6,9})
' or av.Run("SIS.ReturnLSegAzimuth",{1@1,6@9})
'-----
if (SELF.Count = 4) then
  x1 = SELF.Get(0) 'xcoord of first (origin) point
  y1 = SELF.Get(1) 'ycoord of first (origin) point
  x2 = SELF.Get(2) 'xcoord of second point
  y2 = SELF.Get(3) 'ycoord of second point
  p1 = Point.Make(x1,y1)
  p2 = Point.Make(x2,y2)

```

```

elseif (SELF.Count = 2) then
    p1 = SELF.Get(0)           'first (origin) point
    p2 = SELF.Get(1)           'second point
    x1 = p1.GetX
    y1 = p1.GetY
    x2 = p2.GetX
    y2 = p2.GetY
else
    return nil
end
'-----
'calculate the angle using simple trig:
'-----
h = p1.Distance(p2)           'h = the distance between the two
points (hypotenuse)
dX = x2 - x1                   'dX = difference in xcoords
dY = y2 - y1                   'dY = difference in ycoords

a = ((dX / h).ACos.AsDegrees) 'a = the angle made by the two points (angle
clockwise from North in degrees)

if (dY < 0) then
    a = 360 - a
end
'-----
'return the angle as a number between 0 and 360:
'-----
return a

'----Scriptende----

' Name:   SIS.RtDef
'
' Zweck: Definieren einer Route mit 2 Bezugspunkten (verbesserte Version
mit Segmentordnung)
'
'
' Thema:  SIS, Dynamische Sementierung
'
' Voraussetzung: Das aktive Dokument muss ein View sein.
'                Die Elemente (2 Bezugspunkte und mindestens eine Kante)
'                müssen im BP-Thema und im Routenthema bereits selektiert
'                sein.
'                SIS.ListSub wird zur Innenknotenerkennung aufgerufen
'                SIS.LnDefine.Read wird für die Themeninformation
aufgerufen
'                SIS.FindEdge wird zur Kantenauswahl benötigt
'
' Self:
'
' Rückgabe:
'
' Autor:  Horst Baumgartner 2002/01/22
'
t = "SIS-Route"

theProject = av.GetProject
theView = av.GetActiveDoc
theDisplay = theView.GetDisplay

' Fehlerbehandlung:
'
' bei Fehlern die nach Schreibzugriffen erfolgen
' wird RSave auf False gesetzt und die Änderungen durch Recovery

```

```

' nicht gespeichert

RSave = True

ErrMsg = {
"Es müssen mindestens 2 Bezugspunkte ausgewählt werden!",
"Keine Straßenkanten ausgewählt!",
"Bezugspunkte wurden falsch zugeordnet!",
"Die Route ist nicht geschlossen oder enthält eine Kreuzung!",
"Vor Aufruf dieser Funktion muss im Menü SIS ein Linienthema definiert
werden!",
"Fehler in der Datenstruktur!"
}

' Parameter für die verwendeten Themen holen

theLTDefList = av.Run("SIS.LnDefine.Read",0)

if(theLTDefList = nil) then
  MsgBox.Error(ErrMsg.Get(4), t)
  exit
end

theBPTheme = theView.FindTheme(theLTDefList.Get(0))

' Nicht mehr aktuell -> neu definieren

if(theBPTheme = nil) then
  MsgBox.Error(ErrMsg.Get(4), t)
  exit
end

theSrcName = theBPTheme.GetSrcName
theBPFTab = theSrcName.GetEventTable
theBPBitmap = theBPFTab.GetSelection

' Check ob 2 Bezugspunkte vorhanden sind, wenn nicht Abbruch

if(theBPBitmap.Count <> 2) then

  ' Wenn die Bezugspunkte fehlen, dann RVS-Version aufrufen

  if(theBPBitmap.Count = 0) then
    av.run("SIS.RtrVSDef",nil)
    exit
  end
  MsgBox.Error(ErrMsg.Get(0),t)
  exit
end

' Die Ftab des Routen(Kanten)themas holen

theKFtheme = theView.FindTheme(theLTDefList.Get(1))

' Nicht mehr aktuell -> neu definieren

if(theKFtheme = nil) then
  MsgBox.Error(ErrMsg.Get(4), t)
  exit
end

theKFtab = theKFtheme.GetFTab
theKBitmap = theKFtab.GetSelection

' Kantenanzahl ermitteln

```

```

theKAnz = theKBitmap.Count

if(theKAnz = 0) then
  MsgBox.Error(ErrMsg.Get(1),t)
  exit
end

' Nach den Routen und Segmenttabellen suchen
theRDoc = theProject.FindDoc(theLTDefList.Get(3))
theSDoc = theProject.FindDoc(theLTDefList.Get(2))

' Nicht mehr aktuell -> neu definieren

if((theRDoc = nil) or (theSDoc = nil)) then
  MsgBox.Error(ErrMsg.Get(4), t)
  exit
end

' Vtabs der Routen und Segementtabellen

theRVTab = theRDoc.GetVtab
theSVTab = theSDoc.GetVtab

' Bezugspunkte holen und die Struktur in einer Liste
' mit folgenden Format speichern
' 0 BPID
' 1 Stat
' 2 Str_id
' 3 Info

BpList = List.Make

for each prec in theBPBitmap
  BpList.Add({theBPFTab.ReturnValue(theBPFTab.FindField("Pid"),prec),
             theBPFTab.ReturnValue(theBPFTab.FindField("Stat"),prec),
             theBPFTab.ReturnValue(theBPFTab.FindField("Str_Id"),prec),
             theBPFTab.ReturnValue(theBPFTab.FindField("Info"),prec)})
end

ret = MsgBox.ListAsString(
  {BpList.Get(0).Get(0).AsString+";"+BpList.Get(0).Get(3).AsString,
   BpList.Get(1).Get(0).AsString+";"+BpList.Get(1).Get(3).AsString},
  "Startpunkt auswählen",t)

' Benutzerabbruch oder Auswertung

if(ret = nil) then
  exit
else
  if(ret.AsTokens(";").get(0).AsNumber = BpList.Get(0).Get(0)) then
    BPStartIx = 0
    BPEndIx = 1
  else
    BPStartIx = 1
    BPEndIx = 0
  end
end

' Kanten holen, einen gerichteten Graphen mit
' der folgenden Listenstruktur sowie eine Knotenliste dazu aufbauen
' 0 StrID
' 1 FromNode
' 2 ToNode

```

```

' 3 Length
' 4 Direction

theKList= List.Make
theNodeList = List.Make
kcnt = 0
KStartIx = nil
KEndIx = nil

for each krec in theKBitmap
    FromNode = theKFTab.ReturnValue(theKFTab.FindField("FromNode"),krec)
    ToNode = theKFTab.ReturnValue(theKFTab.FindField("ToNode"),krec)
    KStrID = theKFTab.ReturnValue(theKFTab.FindField("Id"),krec)
    theKList.Add({KStrID, FromNode, ToNode,
        theKFTab.ReturnValue(theKFTab.FindField("Length"),krec),1})
    theKList.Add({KStrID, ToNode, FromNode,
        theKFTab.ReturnValue(theKFTab.FindField("Length"),krec),-1})
    ' Start und Endkanten merken
    if(KStrID = BPList.Get(BPStartIx).Get(2)) then
        KStartIx = kcnt
    end
    if(KStrID = BPList.Get(BPEndIx).Get(2)) then
        KEndIx = kcnt
    end
    theNodeList.Add(FromNode)
    theNodeList.Add(ToNode)
    kcnt = kcnt + 2
end

' Wenn die Start- oder die Endkante fehlen, dann Abbruch

if((KStartIx = nil) or (KEndIx = nil)) then
    MsgBox.Error(ErrMsg.Get(2),t)
    exit
end

' Innenknoten feststellen

theHlpNodeList = theNodeList.Clone
theHlpNodeList.RemoveDuplicates

theINodeList = av.Run("SIS.ListSub",{theNodeList,theHlpNodeList})

'FehlerCheck: Anzahl der Innenknoten = Anzahl der Kanten - 1

if(theINodeList.Count < (theKAnz -1)) then
    MsgBox.Error(ErrMsg.get(3),t)
    exit
end

' Falls die Start oder Endkante nur Innenknoten aufweisen, muss
' die Orientierung vom Benutzer angegeben ,und der Graph
' bezüglich der Orientierung so angepasst werden, dass die Reihenfolge
' richtig ermittelt werden kann
' Behandlung der Startkante

if((theINodeList.FindbyValue(theKList.Get(kStartIX).Get(1)) <> -1) and
    (theINodeList.FindbyValue(theKList.Get(kStartIX).Get(2)) <> -1)) then
    if(MsgBox.YesNo("Verläuft die Richtung der Startkante in Richtung der
    neuen Route ?",t,true)) then
        theKList.Get(kStartIX).Set(1,-1)
        theKList.Get(kStartIX+1).Set(2,-1)
    else
        theKList.Get(kStartIX).Set(2,-1)
        theKList.Get(kStartIX+1).Set(1,-1)
    end
end
end

```

```

'Behandlung der Endkante

if((theINodeList.FindbyValue(theKList.Get(kEndIX).Get(1)) <> -1) and
  (theINodeList.FindbyValue(theKList.Get(kEndIX).Get(2)) <> -1)) then
  if(MsgBox.YesNo("Verläuft die Richtung der Endkante in Richtung der
    neuen Route ?",t,true)) then
    theKList.Get(kEndIX).Set(2,-1)
    theKList.Get(kEndIX+1).Set(1,-1)
  else
    theKList.Get(kEndIX).Set(1,-1)
    theKList.Get(kEndIX+1).Set(2,-1)
  end
end
end

RtInfo = MsgBox.Input("Routeninformation",t,"")

' BenutzerAbbruch

if(RtInfo = nil) then
  exit
end

' Felder in der Routentabelle
theRtIdField = theRVTab.FindField("RId")
theRtBPFfromField = theRVTab.FindField("BPFfrom")
theRtBPToField = theRVTab.FindField("BPTo")
theRtLenField = theRVTab.FindField("RLen")
theRTInfoField = theRVTab.FindField("Info")
theRTUserField = theRVTab.FindField("User")
theRTDateField = theRVTab.FindField("Date")

' Neue Routen und SegmentIDs ermitteln

theRID = av.Run("SIS.TabIndex",{theRVtab,theRtIdField})
theSID = av.Run("SIS.TabIndex",{theSVtab,theSVTab.FindField("SegID")})

' Routentabelle und Segementtabelle zum Editieren öffnen

theRVTab.StartEditingWithRecovery
theSVTab.StartEditingWithRecovery

' Datensatz zur Routentabelle hinzufügen

rrec = theRVTab.AddRecord
theRVTab.SetValue(theRtIdField,rrec,theRID)
theRVTab.SetValue(theRtBPFfromField,rrec,BpList.Get(BPStartIx).Get(0))
theRVTab.SetValue(theRtBPToField,rrec,BpList.Get(BPEndIx).Get(0))
theRVTab.SetValue(theRtInfoField,rrec,RtInfo)
theRVTab.SetValue(theRtUserField,rrec,_theUser)
theRVTab.SetValue(theRtDateField,rrec,Date.Now)

' falls der Graph nur aus einer Kante besteht können Orientierung und
' Stationierung direkt von den Bezugspunkten bestimmt werden

if(theKAnz = 1) then
  srec = theSVTab.AddRecord
  theSVTab.SetValue(theSVTab.FindField("Segid"),srec,theSID)
  theSVTab.SetValue(theSVTab.FindField("Rid"),srec,theRID)

  theSVTab.SetValue(theSVTab.FindField("StrId"),srec,BpList.Get(BPStartIx).
    Get(2))
  theSVTab.SetValue(theSVTab.FindField("Order"),srec,0)
  theSVTab.SetValue(theSVTab.FindField("User"),srec,_theUser)
  theSVTab.SetValue(theSVTab.FindField("Date"),srec, Date.Now)

```

```

if(BpList.Get(BPStartIx).Get(1) < BpList.Get(BPEndIx).Get(1)) then
  theRLen = BpList.Get(BPEndIx).Get(1) - BpList.Get(BPStartIx).Get(1)

  theSVTab.SetValue(theSVTab.FindField("Statfrom"),srec,BpList.Get(BPStar
tIx).Get(1))

  theSVTab.SetValue(theSVTab.FindField("Statto"),srec,BpList.Get(BPEndIx)
.Get(1))
  theSVTab.SetValue(theSVTab.FindField("Direction"),srec,1)
else
  theRLen = BpList.Get(BPStartIx).Get(1) - BpList.Get(BPEndIx).Get(1)

  theSVTab.SetValue(theSVTab.FindField("Statfrom"),srec,BpList.Get(BPEndI
x).Get(1))

  theSVTab.SetValue(theSVTab.FindField("Statto"),srec,BpList.Get(BPStartI
x).Get(1))
  theSVTab.SetValue(theSVTab.FindField("Direction"),srec,-1)
end
theSVTab.SetValue(theSVTab.FindField("MsFrom"),srec,0)
theSVTab.SetValue(theSVTab.FindField("MsTo"),srec,theRLen)
else
  ' Reihenfolge der Kanten bestimmen
  ' Zuerst die richtige Orientierung der Startkante feststellen

  ' Zähler für die Reihenfolge

theOrder = 0

  ' bei nur 2 Kanten ist das zweite automatisch das letzte teilstück,
  ' dann ist diese kante auch bei einer pseudokreuzung zulässig und es
  ' kann das Unterdrücken der Endkante ignoriert werden

if(theKAnz = 2) then
  theEndId = -1
else
  theEndId = BpList.Get(BPEndIx).Get(2)
end

  ' prüfen ob es in Richtung der Kantenorientierung weitergeht

found = av.run("SIS.FindEdge",
{theKList, theKList.get(KstartIx).get(0), theKList.get(KstartIx).get(2),
theEndId})

  ' wenn nicht, dann mit der gewendeten Version beginnen

if(found = nil) then
  KstartIx = KstartIx + 1
  found = av.run("SIS.FindEdge",
  {theKList, theKList.get(KstartIx).get(0),
  theKList.get(KstartIx).get(2), theEndId})
end

  ' Fehler in der Datenstruktur, keinen der Datensätze speichern
  ' Programmabbruch

if(found = nil) then
  MsgBox.Error(ErrMsg.Get(5), t)
  theRVTab.StopEditingWithRecovery(false)
  theSVTab.StopEditingWithRevovery(false)
  exit
end

  ' Sonderbehandlung für die Startkante wg. der Teilstationierung
  ' Segment wird nur dann geschrieben, wenn die ermittelte Länge > 0 ist

```

```

if(theKList.Get(KStartIx).Get(4) = 1) then
  theMess = theKList.Get(KStartIx).Get(3) - BpList.Get(BPStartIx).Get(1)
else
  theMess = BpList.Get(BPStartIx).Get(1)
end

if(theMess > 0.005) then
  srec = theSVTab.AddRecord
  theSVTab.SetValue(theSVTab.FindField("Segid"),srec,theSID)
  theSVTab.SetValue(theSVTab.FindField("Rid"),srec,theRID)

  theSVTab.SetValue(theSVTab.FindField("StrId"),srec,theKList.Get(KStartIx).Get(0))
  theSVTab.SetValue(theSVTab.FindField("Order"),srec,theOrder)
  theSVTab.SetValue(theSVTab.FindField("User"),srec,_theUser)
  theSVTab.SetValue(theSVTab.FindField("Date"),srec, Date.Now)

  if(theKList.Get(KStartIx).Get(4) = 1) then

    theSVTab.SetValue(theSVTab.FindField("Statfrom"),srec,BpList.Get(BPStartIx).Get(1))

    theSVTab.SetValue(theSVTab.FindField("Statto"),srec,theKList.Get(KStartIx).Get(3))
    theSVTab.SetValue(theSVTab.FindField("Direction"),srec,1)
  else
    theSVTab.SetValue(theSVTab.FindField("Statfrom"),srec, 0)

    theSVTab.SetValue(theSVTab.FindField("Statto"),srec,BpList.Get(BPStartIx).Get(1))
    theSVTab.SetValue(theSVTab.FindField("Direction"),srec,-1)
  end
  theSVTab.SetValue(theSVTab.FindField("MsFrom"),srec,0)
  theSVTab.SetValue(theSVTab.FindField("MsTo"),srec,theMess)
  theSID = theSID + 1
  theOrder = theOrder + 1
else
  ' eine Kante weniger
  theKanz = theKanz - 1
end
theIndex = found

while((theIndex <> nil) and (theOrder < (theKanz-1)))
  srec = theSVTab.AddRecord
  theSVTab.SetValue(theSVTab.FindField("Segid"),srec,theSID)
  theSVTab.SetValue(theSVTab.FindField("Rid"),srec,theRID)

  theSVTab.SetValue(theSVTab.FindField("StrId"),srec,theKList.Get(theIndex).Get(0))
  theSVTab.SetValue(theSVTab.FindField("Order"),srec,theOrder)
  theSVTab.SetValue(theSVTab.FindField("User"),srec,_theUser)
  theSVTab.SetValue(theSVTab.FindField("Date"),srec, Date.Now)
  theSVTab.SetValue(theSVTab.FindField("Statfrom"),srec,0)

  theSVTab.SetValue(theSVTab.FindField("Statto"),srec,theKList.Get(theIndex).Get(3))

  theSVTab.SetValue(theSVTab.FindField("Direction"),srec,theKList.Get(theIndex).Get(4))
  theSVTab.SetValue(theSVTab.FindField("MsFrom"),srec,theMess)
  theMess = theMess + theKList.Get(theIndex).Get(3)
  theSVTab.SetValue(theSVTab.FindField("MsTo"),srec,theMess)
  theSID = theSID + 1
  theOrder = theOrder + 1
  ' erst wenn nach der letzten kante gesucht wird, ist die EndId zulässig
  if((theOrder + 1) >= theKanz) then

```

```

    theEndId = -1
end
theIndex = av.run("SIS.FindEdge",
{theKList, theKList.get(theIndex).get(0),
theKList.get(theIndex).get(2), theEndId})
end

' Sonderbehandlung für die letzte Kante
' Wird nur dann geschrieben, wenn die Segmentlänge > 0 ist

if(theKList.Get(theIndex).Get(4) = 1) then
    theLStat = BpList.Get(BPEndIx).Get(1)
else
    theLStat = theKList.Get(theIndex).Get(3) - BpList.Get(BPEndIx).Get(1)
end

if(theLStat > 0.005) then
    srec = theSVTab.AddRecord
    theSVTab.SetValue(theSVTab.FindField("Segid"),srec,theSID)
    theSVTab.SetValue(theSVTab.FindField("Rid"),srec,theRID)

    theSVTab.SetValue(theSVTab.FindField("StrId"),srec,theKList.Get(theIndex).Get(0))
    theSVTab.SetValue(theSVTab.FindField("Order"),srec,theOrder)
    theSVTab.SetValue(theSVTab.FindField("User"),srec,_theUser)
    theSVTab.SetValue(theSVTab.FindField("Date"),srec, Date.Now)
    theSVTab.SetValue(theSVTab.FindField("MsFrom"),srec,theMess)

    if(theKList.Get(theIndex).Get(4) = 1) then
        theSVTab.SetValue(theSVTab.FindField("Statfrom"),srec, 0)
        theSVTab.SetValue(theSVTab.FindField("Statto"),srec,
        BpList.Get(BPEndIx).Get(1))
        theSVTab.SetValue(theSVTab.FindField("Direction"),srec,1)
    else
        theSVTab.SetValue(theSVTab.FindField("Statfrom"),srec,
        BpList.Get(BPEndIx).Get(1))
        theSVTab.SetValue(theSVTab.FindField("Statto"),srec,
        theKList.Get(theIndex).Get(3))
        theSVTab.SetValue(theSVTab.FindField("Direction"),srec, -1)
    end
    theMess = theMess + theLStat
    theSVTab.SetValue(theSVTab.FindField("MsTo"),srec, theMess)
end
theRLen = theMess
end

' Routenlänge speichern

theRVTab.SetValue(theRtLenField,rrec,theRLen)

' Routen und Segementtabelle wieder schließen

theRVTab.StopEditingWithRecovery(true)
theSVTab.StopEditingWithRecovery(true)
theRVTab.Flush
theSVTab.Flush

'-----Scriptende-----

' Name:   SIS.RtRVSTDef
'
' Zweck: Definieren einer Route ohne Bezugspunkte (mit Segmentordnung)
'
'
'
' Thema: SIS, Dynamische Sementierung

```

```

'
' Voraussetzung: Das aktive Dokument muss ein View sein.
'               Die Elemente (mindestens eine Kante)
'               müssen im Routenthema bereits selektiert
'               sein.
'               SIS.ListSub wird zur Innenknotenerkennung aufgerufen
'               SIS.LnDefine.Read wird für die Themeninformation
aufgerufen
'               SIS.FindEdge wird zur Kantenauswahl benötigt
'
' Self:
'
' Rückgabe:
'
' Autor: Horst Baumgartner 2002/01/01
'
t = "SIS-Route"

theProject = av.GetProject
theView = av.GetActiveDoc
theDisplay = theView.GetDisplay

' Fehlerbehandlung:
'
' bei Fehlern die nach Schreibzugriffen erfolgen
' wird RSave auf False gesetzt und die Änderungen durch Recovery
' nicht gespeichert

RSave = True

ErrMsg = {
"Keine Straßenkanten ausgewählt!",
"Die Route ist nicht geschlossen oder enthält eine Kreuzung!",
"Vor Aufruf dieser Funktion muss im Menü SIS ein Linienthema definiert
werden!",
"Fehler in der Datenstruktur!"
}

' Parameter für die verwendeten Themen holen

theLTDefList = av.Run("SIS.LnDefine.Read",0)

if(theLTDefList = nil) then
  MsgBox.Error(ErrMsg.Get(2), t)
  exit
end

' Die Ftab des Routen(Kanten)themas holen

theKFtheme = theView.FindTheme(theLTDefList.Get(1))

' Nicht mehr aktuell -> neu definieren

if(theKFtheme = nil) then
  MsgBox.Error(ErrMsg.Get(2), t)
  exit
end

theKFtab = theKFtheme.GetFTab
theKBitmap = theKFtab.GetSelection

' Kantenzahl ermitteln

theKAnz = theKBitmap.Count

```

```

if(theKAnz = 0) then
  MsgBox.Error(ErrMsg.Get(0),t)
  exit
end

' Nach den Routen und Segmenttabellen suchen
theRDoc = theProject.FindDoc(theLTDefList.Get(3))
theSDoc = theProject.FindDoc(theLTDefList.Get(2))

' Nicht mehr aktuell -> neu definieren

if((theRDoc = nil) or (theSDoc = nil)) then
  MsgBox.Error(ErrMsg.Get(2), t)
  exit
end

' Vtabs der Routen und Segementtabellen

theRVTab = theRDoc.GetVtab
theSVTab = theSDoc.GetVtab

' Kanten holen, einen gerichteten Graphen mit
' der folgenden Listenstruktur sowie eine Knotenliste dazu aufbauen
' 0 StrID
' 1 FromNode
' 2 ToNode
' 3 Length
' 4 Direction

theKList= List.Make
theNodeList = List.Make
KStartIx = nil
KEndIx = nil
KCount = 0

for each krec in theKBitmap
  FromNode = theKFtab.ReturnValue(theKFtab.FindField("FromNode"),krec)
  ToNode = theKFtab.ReturnValue(theKFtab.FindField("ToNode"),krec)

  ' Im Falle einer geschlossenen Kante wird diese zuerst einmal
  ' als Endkante festgelegt. Durch den Umstand, dass es nur einen Außen-
  ' knoten gibt, erscheint die Kante mit diesem Knoten automatisch als
  ' Startkante.

  if(FromNode = ToNode) then
    KEndIx = KCount
  end

  KStrID = theKFtab.ReturnValue(theKFtab.FindField("Id"),krec)
  theKList.Add({KStrID, FromNode, ToNode,
    theKFtab.ReturnValue(theKFtab.FindField("Length"),krec),1})
  theKList.Add({KStrID, ToNode, FromNode,
    theKFtab.ReturnValue(theKFtab.FindField("Length"),krec),-1})
  theNodeList.Add(FromNode)
  theNodeList.Add(ToNode)
  KCount = KCount + 1
end

' Innenknoten feststellen

theHlpNodeList = theNodeList.Clone
theHlpNodeList.RemoveDuplicates

theINodeList = av.Run("SIS.ListSub",{theNodeList,theHlpNodeList})

```

```

'FehlerCheck: Anzahl der Innenknoten = Anzahl der Kanten - 1

if(theINodeList.Count < (theKanz -1)) then
  MsgBox.Error(ErrMsg.get(1),t)
  exit
end

' Endpunkte bestimmen

theENodeList = theNodeList - theINodeList

' Endkanten suchen

kStartIx = av.run("SIS.FindEdge",
  {theKList, -1, theENodeList.get(0), -1})

' Nur wenn die Endkante kein Ring ist

if(theENodeList.Count > 1) then
  kEndIx = av.run("SIS.FindEdge",
    {theKList, -1, theENodeList.get(1), -1})
end

if((kStartIx = nil) or (kEndIx = nil)) then
  MsgBox.Error("Start- oder Endkante wurde nicht gefunden.",t)
  exit
end

' nur bei mehr als einer Kante, ohne Ring

if(theKanz > 1) then
  ret = MsgBox.ListAsString(
    {theKList.Get(kStartIx).Get(0).AsString,
     theKList.Get(kEndIx).Get(0).AsString},
    "Startkante auswählen",t)

  ' Benutzerabbruch oder Auswertung

  if(ret = nil) then
    exit
  else
    if(ret.AsNumber = theKList.Get(kEndIx).Get(0)) then
      kSave = kEndIx
      kEndIx = kStartIx
      kStartIx = kSave
    end
  end
end

' Falls die Start oder Endkante nur Innenknoten aufweisen, muss
' die Orientierung vom Benutzer angegeben ,und der Graph
' bezüglich der Orientierung so angepasst werden, dass die Reihenfolge
' richtig ermittelt werden kann
' Behandlung der Startkante

if((theINodeList.FindbyValue(theKList.Get(kStartIX).Get(1)) <> -1) and
  (theINodeList.FindbyValue(theKList.Get(kStartIX).Get(2)) <> -1)) then
  if(MsgBox.YesNo("Verläuft die Richtung der Startkante in Richtung der
  neuen Route ?",t,true)) then
    theKList.Get(kStartIX).Set(1,-1)
    theKList.Get(kStartIX+1).Set(2,-1)
  else
    theKList.Get(kStartIX).Set(2,-1)
    theKList.Get(kStartIX+1).Set(1,-1)
  end
end

```

```

end

'Behandlung der Endkante

if((theINodeList.FindbyValue(theKList.Get(kEndIX).Get(1)) <> -1) and
  (theINodeList.FindbyValue(theKList.Get(kEndIX).Get(2)) <> -1)) then
  if(MsgBox.YesNo("Verläuft die Richtung der Endkante in Richtung der
neuen Route ?",t,true)) then
    theKList.Get(kEndIX).Set(2,-1)
    theKList.Get(kEndIX+1).Set(1,-1)
  else
    theKList.Get(kEndIX).Set(1,-1)
    theKList.Get(kEndIX+1).Set(2,-1)
  end
end
end

RtInfo = MsgBox.Input("Routeninformation",t,"")

' BenutzerAbbruch

if(RtInfo = nil) then
  exit
end

' Felder in der Routentabelle
theRtIdField = theRVTab.FindField("RId")
theRtBPFfromField = theRVTab.FindField("BPFfrom")
theRtBPToField = theRVTab.FindField("BPTo")
theRtLenField = theRVTab.FindField("RLen")
theRtInfoField = theRVTab.FindField("Info")
theRtUserField = theRVTab.FindField("User")
theRtDateField = theRVTab.FindField("Date")

' Neue Routen und SegmentIDs ermitteln

theRID = av.Run("SIS.TabIndex",{theRVtab,theRtIdField})
theSID = av.Run("SIS.TabIndex",{theSVtab,theSVTab.FindField("SegID")})

' Routentabelle und Segementtabelle zum Editieren öffnen

theRVTab.StartEditingWithRecovery
theSVTab.StartEditingWithRecovery

' Datensatz zur Routentabelle hinzufügen

rrec = theRVTab.AddRecord
theRVTab.SetValue(theRtIdField,rrec,theRID)
theRVTab.SetValue(theRtBPFfromField,rrec,-1)
theRVTab.SetValue(theRtBPToField,rrec,-1)
theRVTab.SetValue(theRtInfoField,rrec,RtInfo)
theRVTab.SetValue(theRtUserField,rrec,_theUser)
theRVTab.SetValue(theRtDateField,rrec,Date.Now)

' falls der Graph nur aus einer Kante besteht können Orientierung und
' Stationierung direkt von den Bezugspunkten bestimmt werden

if(theKAnz = 1) then
  if(MsgBox.YesNo("Verläuft die Richtung der Kante in Richtung der neuen
Route ?",t,true)) then
    OneKDir = 1
  else
    OneKDir = -1
  end
end
srec = theSVTab.AddRecord
theSVTab.SetValue(theSVTab.FindField("Segid"),srec,theSID)
theSVTab.SetValue(theSVTab.FindField("Rid"),srec,theRID)

```

```

theSVTab.SetValue(theSVTab.FindField("StrId"),srec,theKList.Get(0).Get(0)
)
theSVTab.SetValue(theSVTab.FindField("Order"),srec,0)
theSVTab.SetValue(theSVTab.FindField("User"),srec,_theUser)
theSVTab.SetValue(theSVTab.FindField("Date"),srec, Date.Now)

theRLen = theKList.Get(0).Get(3)
theSVTab.SetValue(theSVTab.FindField("Statfrom"),srec,0)
theSVTab.SetValue(theSVTab.FindField("Statto"),srec,theRLen)
theSVTab.SetValue(theSVTab.FindField("Direction"),srec,OneKDir)
theSVTab.SetValue(theSVTab.FindField("MsFrom"),srec,0)
theSVTab.SetValue(theSVTab.FindField("MsTo"),srec,theRLen)
else
' Reihenfolge der Kanten bestimmen
' Zuerst die richtige Orientierung der Startkante feststellen

' Zähler für die Reihenfolge

theOrder = 0

' prüfen ob es in Richtung der Kantensorientierung weitergeht

found = av.run("SIS.FindEdge",
{theKList, theKList.get(KstartIx).get(0), theKList.get(KstartIx).get(2),
-1})

' wenn nicht, dann mit der gewendeten Version beginnen

if(found = nil) then
  KstartIx = KstartIx + 1
  found = av.run("SIS.FindEdge",
  {theKList, theKList.get(KstartIx).get(0),
  theKList.get(KstartIx).get(2), -1})
end

' Fehler in der Datenstruktur, keinen der Datensätze speichern
' Programmabbruch

if(found = nil) then
  MsgBox.Error(ErrMsg.Get(3), t)
  theRVTab.StopEditingWithRecovery(false)
  theSVTab.StopEditingWithRevoveroy(false)
  exit
end

srec = theSVTab.AddRecord
theSVTab.SetValue(theSVTab.FindField("Segid"),srec,theSID)
theSVTab.SetValue(theSVTab.FindField("Rid"),srec,theRID)

theSVTab.SetValue(theSVTab.FindField("StrId"),srec,theKList.Get(KstartIx)
.Get(0))
theSVTab.SetValue(theSVTab.FindField("Order"),srec,theOrder)
theSVTab.SetValue(theSVTab.FindField("User"),srec,_theUser)
theSVTab.SetValue(theSVTab.FindField("Date"),srec, Date.Now)

theSVTab.SetValue(theSVTab.FindField("Statfrom"),srec,0)

theSVTab.SetValue(theSVTab.FindField("Statto"),srec,theKList.Get(KstartIx)
.Get(3))

if(theKList.Get(KstartIx).Get(4) = 1) then
  theSVTab.SetValue(theSVTab.FindField("Direction"),srec,1)
else
  theSVTab.SetValue(theSVTab.FindField("Direction"),srec,-1)

```

```

end

theMess = theKList.Get(KStartIx).Get(3)

theSVTab.SetValue(theSVTab.FindField("MsFrom"),srec,0)
theSVTab.SetValue(theSVTab.FindField("MsTo"),srec,theMess)
theSID = theSID + 1
theOrder = theOrder + 1

theIndex = found

while((theIndex <> nil) and (theOrder <= (theKAnz-1)))
  srec = theSVTab.AddRecord
  theSVTab.SetValue(theSVTab.FindField("Segid"),srec,theSID)
  theSVTab.SetValue(theSVTab.FindField("Rid"),srec,theRID)

  theSVTab.SetValue(theSVTab.FindField("StrId"),srec,theKList.Get(theIndex).Get(0))
  theSVTab.SetValue(theSVTab.FindField("Order"),srec,theOrder)
  theSVTab.SetValue(theSVTab.FindField("User"),srec,_theUser)
  theSVTab.SetValue(theSVTab.FindField("Date"),srec, Date.Now)
  theSVTab.SetValue(theSVTab.FindField("Statfrom"),srec,0)

  theSVTab.SetValue(theSVTab.FindField("Statto"),srec,theKList.Get(theIndex).Get(3))

  theSVTab.SetValue(theSVTab.FindField("Direction"),srec,theKList.Get(theIndex).Get(4))
  theSVTab.SetValue(theSVTab.FindField("MsFrom"),srec,theMess)
  theMess = theMess + theKList.Get(theIndex).Get(3)
  theSVTab.SetValue(theSVTab.FindField("MsTo"),srec,theMess)
  theSID = theSID + 1
  theOrder = theOrder + 1
  theIndex = av.run("SIS.FindEdge",
    {theKList, theKList.get(theIndex).get(0),
    theKList.get(theIndex).get(2), -1})
end

  theRlen = theMess
end

' Routenlänge speichern

theRVTab.SetValue(theRtLenField,rrec,theRlen)

' Routen und Segementtabelle wieder schließen

theRVTab.StopEditingWithRecovery(true)
theSVTab.StopEditingWithRecovery(true)
theRVTab.Flush
theSVTab.Flush

'-----Scriptende-----

' Name:   SIS.RtMkTables
'
' Zweck: Erstellen der für ein Routenthema erforderlichen
'        Tabellen
'
'
'
' Thema: SIS, Dynamische Sementierung
'
' Voraussetzung:
'
'

```

```

'
' Self:
'
' Rückgabe: Liste mit den Filenamen als String
'           Element 0: Routentabelle
'           Element 1: Segmenttabelle
'
' Autor: Horst Baumgartner 2000/02/19
'

theRouteTabFName =
FileDialog.Put("URoute.dbf".AsFilename,"*.*","Routentabelle")

' Abbruch wenn eine der Tabellen nicht definiert sein sollte

if(theRouteTabFName = nil) then
    exit
end

theSegTabFName =
FileDialog.Put("USeg.dbf".AsFilename,"*.*","Segmenttabelle")

if(theSegTabFName = nil) then
    exit
end

'Namen in Liste für die Rückgabeliste

ret = {theRouteTabFName.AsString,theSegTabFName.AsString}

theRVTab = VTab.MakeNew(theRouteTabFName,dbase)
theSVTab = VTab.MakeNew(theSegTabFName,dbase)

theRTable = Table.Make(theRVTab)
theSTable = Table.Make(theSVTab)

theRTable.SetName(theRouteTabFName.GetBaseName)
theSTable.SetName(theSegTabFName.GetBaseName)

' Felder für die Routentabelle

Rid = Field.Make("Rid",#FIELD_DOUBLE,16,0)
BPFfrom = Field.Make("BPFfrom",#FIELD_DOUBLE,16,0)
BPTo = Field.Make("BPTo",#FIELD_DOUBLE,16,0)
Rlen = Field.Make("Rlen",#FIELD_DOUBLE,16,2)
RInfo =Field.Make("Info",#FIELD_CHAR,30,0)
RUser =Field.Make("User",#FIELD_CHAR,10,0)
RDate =Field.Make("Date",#FIELD_DATE,8,0)

theRVTab.AddFields({Rid,BPFfrom,BPTo,Rlen,RInfo,RUser,RDate})

'Felder für die Liniensegmentierung

SegId = Field.Make("SegId",#FIELD_DOUBLE,16,0)
Rid = Field.Make("Rid",#FIELD_DOUBLE,16,0)
StrId = Field.Make("StrId",#FIELD_DOUBLE,16,0)
StatFrom = Field.Make("StatFrom",#FIELD_DOUBLE,16,2)
StatTo = Field.Make("StatTo",#FIELD_DOUBLE,16,2)
SegOrder = Field.Make("Order",#FIELD_DOUBLE,16,0)
SegDir = Field.Make("Direction",#FIELD_DOUBLE,2,0)
SegMsFrom = Field.Make("MsFrom",#FIELD_DOUBLE,16,2)
SegMsTo = Field.Make("MsTo",#FIELD_DOUBLE,16,2)
SegUser =Field.Make("User",#FIELD_CHAR,10,0)
SegDate =Field.Make("Date",#FIELD_DATE,8,0)

```

```

theSVTab.AddFields({SegId,Rid,StrId,StatFrom,StatTo,SegOrder,SegDir,SegMsFrom,
                    SegMsTo, SegUser, SegDate})

' Editieren beenden

theRVtab.SetEditable(false)
theSVtab.SetEditable(false)
theRVTab.Flush
theSVTab.Flush

'-----Scriptende-----

' Name:   SIS.RtRemove
' Zweck: Entfernen einer Route nach Selektion mit einem Punkt
'         Datensätze werden aus der Segment- und Routentabelle gelöscht
'
'
' Thema:  SIS, Dynamische Sementierung
'
' Voraussetzung: Das aktive Dokument muss ein View sein.
'                 Routine muß an ein Tool gehängt werden
'                 SIS.LnDefine.Read wird für die Themeninformation
aufgerufen
'                 SIS.IsLine.Update ist als Updatefunktion im Tool zu
'                 integrieren
'
' Self:
'
' Rückgabe:
'
' Autor:  Horst Baumgartner 2000/03/03
'
'

t = "SIS-Route"

theProject = av.GetProject
theView = av.GetActiveDoc
theDisplay = theView.GetDisplay

theTheme = theView.getActiveThemes.Get(0)

' Parameter für die verwendeten Themen holen

theLTDefList = av.Run("SIS.LnDefine.Read",0)

if(theLTDefList = nil) then
  MsgBox.Error("Vor Aufruf dieser Funktion muss im Menü SIS ein Linienthema
  definiert werden!", t)
  exit
end

' Nach den Routen und Segmenttabellen suchen
theRDoc = theProject.FindDoc(theLTDefList.Get(3))
theSDoc = theProject.FindDoc(theLTDefList.Get(2))

' Nicht mehr aktuell -> neu definieren

if((theRDoc = nil) or (theSDoc = nil)) then
  MsgBox.Error("Vor Aufruf dieser Funktion muss im Menü SIS ein Linienthema
  definiert werden!", t)
  exit
end

```

```

' Ftab eines Dyn.Segmentierungstheams ist Srcname, abgesichert durch die
' Updateroutine SIS.IsLine.Update

theSrcName = theTheme.GetSrcName
theVTab = theSrcName.GetEventTable

thePoint = theDisplay.ReturnUserPoint
theVTab.SelectByPoint(thePoint,5,#VTAB_SELTYPE_NEW)

theBitmap = theVTab.GetSelection
theSelRID = theVTab.FindField("RId")

' keine Route gewählt

if(theBitmap.Count = 0) then
    exit
end

rec = theBitmap.getNextSet(-1)

if(theBitmap.Count > 1) then
    theSelList = List.Make
    for each rec in theBitmap
        theSelList.Add(theVTab.ReturnValue(theSelRID,rec).AsString)
    end
    RIDString = MsgBox.ListAsString(theSelList,"Welche Route soll gelöscht
    werden:", "SIS-Route")
else
    RIDString = theVTab.ReturnValue(theSelRID,rec).AsString
end

if(RIDString = Nil) then
    exit
end

if(MsgBox.LongYesNo("Soll RID:++RIDString ++ "wirklich gelöscht
werden?","Info",true)=false) then
    exit
end

' Ab hier wird gelöscht

theRVTab = theRDoc.GetVtab
theSVTab = theSDoc.GetVtab
theRBitmap = theRVTab.GetSelection
theSBitmap = theSVTab.GetSelection

' Suchbegriff festlegen

expr = "([RId] =" + RIDString +)"

' Datensätze in der Routen- und Segemnt-Tabelle auswählen

theRVTab.Query(expr,theRBitmap,#VTAB_SELTYPE_NEW)
theSVTab.Query(expr,theSBitmap,#VTAB_SELTYPE_NEW)

if(theRVTab.StartEditingWithRecovery and theSVTab.StartEditingWithRecovery)
then
    theRVTab.BeginTransaction
    theSVTab.BeginTransaction
    theRVTab.RemoveRecords(theRBitmap)
    theSVTab.RemoveRecords(theSBitmap)
    theRVTab.EndTransaction
    theSVTab.EndTransaction
else
    MsgBox.Error("Route kann nicht gelöscht werden!","SIS-Route")
end

```

```

theRVTab.StopEditingWithRecovery(true)
theSVTab.StopEditingWithRecovery(true)
theRVTab.Flush
theSVTab.Flush

```

```
'-----Scriptende-----'
```

```

' Name:   SIS.SetUser
'
' Title:  Benutzererkennung setzen
'
'
' Topics: SIS Werkzeuge
'
' Description: Setzt die globale Variable _theUser
'
'
' Requires:
'
' Self:
'
' Returns:

```

```

_theUser = MsgBox.Input("Benutzererkennung eingeben","SIS","Admin")
if(_theUser = nil) then
  _theUser = "Unknown"
end

```

```
'-----Scriptende-----'
```

```

' Name:   SIS.ShowStreet
'
' Zweck:  Sucht im aktuellen View nach Straßen nach Angabe eines
'         Teilstrings der Straßenbezeichnung
'
' Thema:  SIS, Werkzeuge
'
' Voraussetzung: Das aktive Dokument muss ein View sein.
'               Das Straßenthema muss vor anderen Themen mit dem
Tabellenfeld
'               [StrNum] im View vorliegen. Die Tabelle
'               strverz.dbf mit dem Straßenverzeichnis muss vorhanden
sein.
'
'
' Self:
'
' Rückgabe:
'
' Autor:  Horst Baumgartner 2001/12/16
'

```

```
t="SIS-Straßensuche"
```

```

'-----Tabellenname Straßenverzeichnis---
sVerzName ="strverz.dbf"
'-----

```

```

theView = av.GetActiveDoc
theThemeList = theView.getThemes

```

```

' Alle Themen nach dem Feld StrNum durchsuchen

for each theTheme in theThemeList
  if(theTheme.Is(Ftheme)) then
    theFtab = theTheme.GetFtab
    if((theFtab <> nil) and ((theFtab.GetShapeClass.GetClassName =
      "PolyLine") or
      (theFtab.GetShapeClass.GetClassName = "PolyLineM")))) then
      theStrnumFld = theFtab.FindField("Strnum")
      if(theStrNumFld <> nil) then
        theFTheme = theTheme
        break
      end
    end
  end
end
end

' wenn keines der Themen ein Attribut mit der Bezeichnung Strnum enthält
' dann abbrechen

if(theStrNumFld = nil) then
  MsgBox.Error("Im aktuellen View ist kein Straßenthema enthalten!",t)
  exit
end

' Straßenliste anlegen

theStrList = List.Make

SVerzDoc = av.GetProject.FindDoc(SVerzName)
if(SVerzDoc <> nil) then
  theVTab = SVerzDoc.GetVTab
  theSString = MsgBox.Input("Suchzeichenfolge:",t,"")
  if(theSString = nil) then
    exit
  end
  theStrNamFld = theVTab.FindField("StrName")
  theStrSchlFld = theVTab.FindField("StrSchl")
  ' Fehler
  if((theStrNamFld = nil) or (theStrSchlFld = nil)) then
    MsgBox.Error("Das Straßenverzeichnis hat ein falsches Format!",t)
    exit
  end
  theVBitmap = theVTab.GetSelection
  expr = "[StrName] = "*" + theSString + "*"
  theVTab.Query(expr, theVBitmap, #VTAB_SELTYPE_NEW)
  if(theVBitmap.Count = 0) then
    MsgBox.Info("Es wurde kein Straßenname gefunden, der die angegebene"+
      nl +
      "Zeichenfolge enthält!",t)
    exit
  end
  if(theVBitmap.Count = 1) then
    rec = theVBitmap.GetNextSet(-1)
    theStrNum = theVTab.ReturnValue(theStrSchlFld, rec)
  else
    for each rec in theVBitmap
      theStrList.Add(theVTab.ReturnValue(theStrSchlFld, rec).AsString + ","
        + theVTab.ReturnValue(theStrNamFld, rec))
    end
    theStrNumSel = MsgBox.ListasString(theStrList,"Straße auswählen",t)
    if(theStrNumSel = nil) then exit end
    theStrNum = theStrNumSel.AsTokens(",").get(0).AsNumber
  end
end
else
  MsgBox.Error("Die Tabelle"++SVerzName++"mit dem Straßenverzeichnis konnte
    nicht gefunden werden!",t)

```

```

    exit
end

' Straße im Straßenthema auswählen und anzeigen

expr = "([StrNum] = " + theStrNum.AsString + ")"
theFBitmap = theFtab.GetSelection
theFtab.Query(expr, theFBitmap, #VTAB_SELTYPE_NEW)
if(theFBitmap.Count = 0) then
    MsgBox.Info("Die Straßennummer="++theStrNum.AsString++"wurde nicht
        gefunden.",t)
    exit
end

av.GetProject.SetModified(true)

r = Rect.MakeEmpty

if (theFTheme.CanSelect) then
    r = r.UnionWith(theFTheme.GetSelectedExtent)
end

if (r.IsEmpty) then
    return nil
elseif ( r.ReturnSize = (0@0) ) then
    theView.GetDisplay.PanTo(r.ReturnOrigin)
else
    theView.GetDisplay.SetExtent(r.Scale(1.1))
end

'-----Scriptende-----

' Name:  SIS.Split
' Zweck: Eine durch den Anwender definierte Polylinie teilt den
Straßengraphen und
'         erzeugt eine Kreuzung. Die Originalkante wird dabei nicht
gelöscht, sondern
'         nur als ungültig markiert.
'         Die Attribute der Originalkante werden auf die neu entstehenden
kanten kopiert.
'         Die längere Kante der Anwenderpolylinie wird als Straßensegment
übernommen.
'
' Thema: SIS, Dynamische Sementierung
'
' Voraussetzung: Das aktive Dokument muss ein View sein.
'                 Routine muß an ein Tool gehängt werden
'                 SIS.StrInvalid wird aufgreufen um die Kante als ungültig
'                 zu markieren
'                 Das Thema muss zum Editieren geöffnet sein
'
'
' Self:
'
' Rückgabe:
'
' Autor: Horst Baumgartner 2001/10/13
'
'

theView = av.GetActiveDoc
l = theView.ReturnUserPolyLine
theTheme = theView.GetEditableTheme

if (l.IsNull) then
    return nil

```

```

else
  if (theTheme <> nil) then
    theTheme.GetFtab.BeginTransaction
    theFtab = theTheme.GetFtab
    theField = theFtab.FindField("Shape")
    theLengthFld = theFtab.FindField("Length")
    theStrNumFld = theFtab.FindField("StrNum")
    theGattungFld = theFtab.FindField("Gattung")
    theOnewayFld = theFtab.FindField("Oneway")
    theKatFld = theFtab.FindField("Kat")
    theSpeedFld = theFtab.FindField("Speed")
    theUserFld = theFtab.FindField("User")
    theDateFld = theFtab.FindField("Date")

    theType = theField.GetType
    if ((theType = #FIELD_SHAPEPOLY) or (theType = #FIELD_SHAPELINE)) then
      theTheme.SelectByPolyLine(1,#VTAB_SELTYPE_NEW)

      ' zuerst alle von der Polylinie geschnittenen Kanten als ungültig
      markieren

      av.run("SIS.StrInvalid",nil)

      theBitmap = theFtab.GetSelection

      ' bereits inaktiv gesetzte Kanten sind aus der Selektion zu entfernen
      ' nämlich dann, wenn diese aus der aktuellen Editiersession
      entstammen
      ' BUG - Abfrage wird nicht akzeptiert !!!!

      ' theFtab.Query("[Invalid] <= 0.AsDate" ,theBitmap,
      #VTAB_SELTYPE_AND)
      ' theBitmap = theFtab.GetSelection

      ' Kreuzungen einfügen und Attribute kopieren
      ' in theMergedPoly werden alle selektierten Polylinien aus dem
      Straßengraphen
      ' zusammengefaßt, um die Schnittmenge mit der neuen Straßenkante
      bilden zu können

      theMergedPoly = PolyLine.Make({{}})

      for each rec in theBitmap
        theShape = theFtab.ReturnValue(theField,rec)
        theMergedPoly = theMergedPoly.ReturnMerged(theShape.AsPolyline)
        theShapeList = theShape.Split(1)
        for each pelem in theShapeList
          newrec = theFtab.AddRecord
          theFtab.SetValue(theField,newrec,pelem)
          theFtab.SetValue(theLengthFld,newrec,pelem.ReturnLength)

          theFtab.SetValue(theStrNumFld,newrec,theFtab.ReturnValue(theStrNumFld,rec))

          theFtab.SetValue(theGattungFld,newrec,theFtab.ReturnValue(theGattungFld,rec))

          theFtab.SetValue(theKatFld,newrec,theFtab.ReturnValue(theKatFld,rec))

          theFtab.SetValue(theSpeedFld,newrec,theFtab.ReturnValue(theSpeedFld,rec))

          theFtab.SetValue(theOnewayFld,newrec,theFtab.ReturnValue(theOnewayFld,rec))
          theFtab.SetValue(theUserFld,newrec,_theUser)

```

```

        theFtab.SetValue(theDateFld,newrec,Date.Now)
    end
end

' Kanten mit 10 und mehr Metern bleiben erhalten und werden
gespeichert
'
theShapeList = l.Split(theMergedPoly)
for each pelem in theShapeList
    if(pelem.ReturnLength >= 10) then
        theLen = pelem.ReturnLength
        newrec = theFtab.AddRecord
        theFtab.SetValue(theField,newrec,pelem)
        theFtab.SetValue(theLengthFld,newrec,theLen)
        theFtab.SetValue(theUserFld,newrec,_theUser)
        theFtab.SetValue(theDateFld,newrec,Date.Now)
    end
end

    theTheme.GetFtab.EndTransaction
end
end
av.GetProject.SetModified(true)
theTheme.Invalidate(true)
end

'-----Scriptende-----

' Name:  SIS.SplitUpdate
' Zweck: Tested ob das Thema zum Editieren geöffnet ist.
'       ToolCheck für die Funktionen SIS.Split und
'       SIS.StrInvalidByPoint
'
'
' Thema: SIS, Dynamische Sementierung
'
' Voraussetzung:
'
' Self:
'
' Rückgabe:
'
' Autor: Horst Baumgartner 2001/10/13
'
'

theView = av.GetActiveDoc
theTheme = theView.GetEditableTheme

if (theTheme <> nil) then
    SELF.SetEnabled(true)
    return nil
end

SELF.SetEnabled(false)

'-----Scriptende-----

' Name:  SIS.StrGraphMake
'
' Zweck: Erstellen eines Linienthemas für den
'       Straßengraphen
'
'
'
'

```

```
' Thema: SIS, Dynamische Sementierung
'
' Voraussetzung: SIS.MkFTheme wird aufgerufen
'
'
' Self:
'
' Rückgabe:
'
' Autor: Horst Baumgartner 2001/11/21
'
```

```
theFieldList = {
Field.Make("Strnum",#FIELD_DOUBLE,9,0),
Field.Make("ID",#FIELD_DOUBLE,16,0),
Field.Make("Length",#FIELD_DOUBLE,16,2),
Field.Make("Gattung",#FIELD_DOUBLE,2,0),
Field.Make("Kat",#FIELD_DOUBLE,2,0),
Field.Make("Fromnode",#FIELD_DOUBLE,12,0),
Field.Make("Tonode",#FIELD_DOUBLE,12,0),
Field.Make("Oneway",#FIELD_CHAR,2,0),
Field.Make("Invalid",#FIELD_DATE,8,0),
Field.Make("Speed",#FIELD_DOUBLE,5,0),
Field.Make("Altbez",#FIELD_CHAR,10,0),
Field.Make("Vero",#FIELD_CHAR,30,0),
Field.Make("Komb",#FIELD_CHAR,5,0),
Field.Make("Upass",#FIELD_DOUBLE,3,0),
Field.Make("Seconds",#FIELD_DOUBLE,12,0),
Field.Make("User",#FIELD_CHAR,10,0),
Field.Make("Date",#FIELD_DATE,8,0)
}
```

```
av.Run("SIS.MkFTheme",{ "Straßengraphen
anlegen","strassen.shp",theFieldList, Polyline})
```

```
'-----Scriptende-----
```

```
' Name: SIS.StrInvalid
' Zweck: Setzen des aktuellen Datums für selektierte Elemente im
' Feld Invalid der Ftab im aktuellen Thema
'
'
' Thema: SIS, Dynamische Sementierung
'
' Voraussetzung: Das aktive Dokument muss ein View sein.
' Ein Thema muss aktiv sein.
' Das Feld Invalid muss in der Ftab vorhanden sein.
' Mindestens ein Element muss selektiert sein.
'
' Self:
'
' Rückgabe:
'
' Autor: Horst Baumgartner 2001/10/13
'
```

```
t = "SIS-Element deaktivieren"
```

```
theProject = av.GetProject
theView = av.GetActiveDoc
```

```
theTheme = theView.getActiveThemes.Get(0)
```

```

' wenn kein Thema aktiv ist, dann Abbruch

if((theTheme = nil) or(theTheme.Is(FTheme).Not)) then
  exit
end

theFtab = theTheme.GetFTab

theInvalidField = theFtab.FindField("Invalid")

if(theFtab.IsEditable)then
  if(theInvalidField = nil) then
    exit
  end

  theDate = Date.Now

  if(theFtab.getSelection.Count = 0)then
    MsgBox.Info("Im aktiven Thema sind keine Elemente ausgewählt!",t)
    exit
  end

  for each rec in theFtab.getSelection
    theFtab.SetValue(theInvalidField, rec, theDate)
  end

else
  MsgBox.Error("Dieses Thema kann nicht editiert werden!",t)
  exit
end

'-----Scriptende-----

' Name: SIS.StrInvalidByPoint
' Zweck: Der Anwender wählt ein Kante durch einen Punkt aus um diese als
ungültig zu
'      kennzeichnen. Die Originalkante wird dabei nicht gelöscht, sondern
'      nur im Feld Invalid als ungültig gekennzeichnet.
'
' Thema: SIS, Dynamische Sementierung
'
' Voraussetzung: Das aktive Dokument muss ein View sein.
'                Routine muß an ein Tool gehängt werden .
'                SIS.StrInvalid wird aufgreufen um die selektierte Kante
als ungültig
'                zu markieren.
'                Das Thema muss zum Editieren geöffnet sein
'
' Self:
'
' Rückgabe:
'
' Autor: Horst Baumgartner 2001/10/17
'
'

theView = av.GetActiveDoc
p = theView.GetDisplay.ReturnUserPoint
theTheme = theView.GetEditableTheme

if (p.IsNull) then
  return nil
else

```

```

if (theTheme <> nil) then
  theTheme.GetFtab.BeginTransaction
  theFtab = theTheme.GetFtab
  theField = theFtab.FindField("Shape")
  theType = theField.GetType
  if ((theType = #FIELD_SHAPEPOLY) or (theType = #FIELD_SHAPELINE)) then
    theTheme.SelectByPoint(p,#VTAB_SELTYPE_NEW)

    ' jene durch den Punkt selektierten Kanten werden auf ungültig
    gesetzt

    av.run("SIS.StrInvalid",nil)
  end
  theTheme.GetFtab.EndTransaction
end
end
end

```

```
'-----Scriptende-----
```

```

' Name:   SIS.StrVerzMkTable
'
' Zweck: Erstellen einer Straßentabelle
'
'
' Thema:  SIS, Dynamische Sementierung
'
' Voraussetzung: SIS.MkTable wird aufgerufen
'
'
' Self:
'
' Rückgabe:
'
' Autor:  Horst Baumgartner 2001/11/21
'
'
' Felder für die Straßentabelle
' StrSchl   StraßenID (Schlüssel)
' StrName   Straßenbezeichnung
' Oestrnr   ÖSTAT-Straßennummer
' Gemnr     Gemeindefnummer
' Gembez    Gemeindebezeichnung
' User      Benutzerkennung
' Date      Datum

```

```

theFieldList = {
Field.Make("Strschl",#FIELD_DOUBLE,5,0),
Field.Make("Strname",#FIELD_CHAR,30,0),
Field.Make("Oestrnr",#FIELD_DOUBLE,6,0),
Field.Make("Gemnr",#FIELD_DOUBLE,5,0),
Field.Make("GemBez",#FIELD_CHAR,20,0),
Field.Make("User",#FIELD_CHAR,10,0),
Field.Make("Date",#FIELD_DATE,8,0)
}

```

```

av.Run("SIS.MkTable",{ "Straßentabelle
anlegen", "strverz.dbf", theFieldList, true})

```

```
'-----Scriptende-----
```

```
' Name:   SIS.TabIndex
'
```

```

' Zweck: Ermittelt die nächste freie Nummer in einem Tabellenfeld
'
' Aufruf mit: theIx = av.Run("SIS.TabIndex",{theVtab,theField})
'
'
' Thema: SIS, Tabellenwerte
'
' Voraussetzung:
'
'
' Self:   VTab
'        Field
'
' Rückgabe: Nummer oder nil im Fehlerfall
'
' Autor: Horst Baumgartner 2000/02/24
'

theVTab = self.Get(0)
theField = self.Get(1)

' Feld nicht existent oder nicht Numerisch

if(theField.IsTypeNumber.Not) then
  return(nil)
end

theIx = 0

for each rec in theVTab
  s = theVTab.ReturnValue(theField,rec)
  if(s > theIx) then
    theIx = s
  end
end

return(theIx + 1)

'----Scriptende----

' Name:   SIS.Table.List
'
' Zweck: Sucht alle zur verfügbaren Tabellen
'        und gibt diese in einer Liste zurück
'
'
' Thema: SIS, Dynamische Sementierung
'
' Voraussetzung:
'
' Self:
'
' Rückgabe: Liste der Tabellennamen
'
' Autor: Horst Baumgartner 2000/02/22
'

theProj = av.GetProject

theDocList = theProj.GetDocs

theTabList = List.Make

for each i in theDocList
  if(i.is(Table)) then

```

```

        theTabList.Add(i.GetName)
    end
end
return(theTabList)

```

```
'-----Scriptende-----
```

```

'
' SIS.Table.SortPhysical
' (c) 1998-2000 Quantitative Decisions, Merion, PA USA
' http://www.quantdec.com
'
' Physically sort records in a table or theme.
'
' SELF:    void
'
' RETURNS: Error code (integer).  See the "constants" section.
'
' CONTEXT: Run from a table or a view with one active ftheme.
'
' All selected records are sorted on the active field
' and moved in sort order to the end of the physical table.
' If no records are selected, the process is applied to all
' records.
' If there is no active field, the user is prompted to select one.
' In any event, the user is asked whether to sort in
' ascending or descending order.
'
' Intended to establish drawing order in editable themes:
' shapes are drawn in the order encountered while processing
' the FTab.  Thus this operation visually "promotes" the
' selected features to the top and they are drawn from first
' (in sort order) to last among themselves.  For example,
' if the selection consists of a single record, it is
' promoted to the top of the drawing stack.
'
' W. Huber, Quantitative Decisions, Merion, PA  USA 6/11/98
' whuber@Quantdec.com
'
' 6/18/98: Added f.IsEditable protection against nasty AV
'          behavior.
' 6/27/98: Modified from Table.PermutePhysical.
'          Most of the error branches have not been tested.
'          Tested with small themes, with and without selections,
'          with and without definitions set.
' 2/22/00: Improved error checking.  Created return codes.
'          Added progress monitoring to long operations.
'          Some optimizations made.
'
' Comments:
' This script is simple-minded about processing themes: it
' processes only the first active theme, if any.  Likewise,
' although it does some error checking, it merely return NILs
' on encountering an error condition.
'
' Sorting is accomplished through the table.sort request.
' Being built in, it's probably the fastest way available,
' and it does not require creating an Avenue list of keys in
' memory.
'
' Dealing with tables is a real bear.  First, to sort them,
' their windows have to be open.  This automatically adds
' them to the project.  Second, as soon as a record is
' added, the sort is broken and the table reverts to physical
' order.  Third, if one asks a table for its active field
' without the table being open, ArcView crashes.
'
'

```

```

' Our work-around is to do all the ConvertRowToRecord
' requests at once and cache their results in a list. Since
' the primary purpose of this script is to reorder features
' and, presumably, most themes won't have that many features
' (less than 10,000 typically), this doesn't create such a
' huge performance hit.
'
' Another aberrant Avenue behavior encountered in testing
' this script is that a selection bitmap can have defined-out
' bits set. Thus, for example, it is possible for the number
' of selected records to exceed the number of visible records.
' This script therefore takes the precaution of anding the
' definition bitmap with the selection bitmap.
'
' ArcView seems to do a good job of packing the underlying
' tables after removing records.
'-----'
'
' Common message box title:
'
sTitle = "Physical Sort"
'
' Defined return codes (constants):
'
nSUCCESS      = 0
nERR_NOTHEME  = -1 ' No active ftheme
nERR_BADFTAB  = -2 ' Problem with the FTab
nERR_CONTEXT  = -3 ' Not a view or table
nERR_NOFIELDS = -4 ' No visible fields for sorting
nERR_USER     = -5 ' User cancellation
nERR_EDIT     = -6 ' Cannot edit the table
nERR_ADD      = -7 ' Cannot add a new record
nERR_SAVE     = -8 ' Cannot save the transactions
'
' If the current document is a table, obtain its VTab.
' Otherwise, use the first active theme's FTab, if there
' is an active theme.
'
aTable = av.GetActiveDoc
if (av.GetActiveDoc.Is(Table)) then
  aVTab = av.GetActiveDoc.GetVTab
  wasAdded = FALSE
elseif (av.GetActiveDoc.Is(View)) then
  lstThemes = {}
  for each theTheme in av.GetActiveDoc.GetActiveThemes
    if (theTheme.Is(FTheme)) then
      lstThemes.Add(theTheme)
    end
  end
  if (lstThemes.Count = 0) then
    MsgBox.Info("You need to activate an FTheme to process.", sTitle)
    return nERR_NOTHEME
  end
  aVTab = lstThemes.Get(0).GetFTab
  aTable = Table.Make(aVTab)
  if (aTable = NIL) then
    MsgBox.Error("It appears this theme's feature table has an error.",
    sTitle)
    return nERR_BADFTAB
  end
  aTable.SetName("Made by"++sTitle)
  wasAdded = TRUE
else
  MsgBox.Info("Run this script from a View or Table document.", sTitle)
  return nERR_CONTEXT
end
'
'

```

```

' Usual housekeeping for processing all records if none are selected.
,
theBitmap = aVTab.GetSelection
theBitmap.And(aVTab.GetDefBitmap)      ' 22 Feb 2000
aVTab.UpdateSelection                  ' 22 Feb 2000
wasUnselected = (theBitmap.Count = 0)
if (wasUnselected) then
  theBitmap.SetAll
end
'
' Establish the sort order. Ask user for sort field, if necessary,
' and whether this is an ascending or descending sort.
,
if (aTable.GetWin.IsOpen.Not) then
  aTable.GetWin.Open
end
fldSort = aTable.GetActiveField '--springs an Avenue bug if the table is
not open
if (fldSort = NIL) then
  lstStrFields = {}
  for each theField in aVTab.GetFields
    if (theField.IsTypeShape.Not and theField.IsVisible) then
      lstStrFields.Add(theField.GetAlias)
    end
  end
  if (lstStrFields.Count = 0) then
    MsgBox.Error("No data fields are visible for sorting--" +
      "select Table|Properties to make them visible;
      aborting...", sTitle)
    if (wasAdded) then
      av.GetProject.RemoveDoc(aTable)
    end
    return nERR_NOFIELDS
  end
  strFldSort = MsgBox.ChoiceAsString(lstStrFields, "Select a field for
  sorting", sTitle)
  if (strFldSort = NIL) then
    return nERR_USER
  end
  fldSort = aVTab.FindField(strFldSort)
end
IsAscending = MsgBox.YesNo("Sort ascending (yes) or descending (no)?",
  sTitle ++ "[" + fldSort.GetAlias + "]", TRUE)
,
' Start editing the VTab.
,
if (aVTab.StartEditingWithRecovery.Not) then
  MsgBox.Error("Unable to edit this table--make sure it's in dbf format, is
  not read-only," ++
    "and that its folder is not read-only; aborting...", sTitle)
  if (wasAdded) then
    av.GetProject.RemoveDoc(aTable)
  end
  return nERR_EDIT
end
,
' Sort the table document.
,
av.ClearStatus
av.ShowMsg("Sorting...")

aTable.Sort(fldSort, IsAscending.Not)
aTable.PromoteSelection
M = aVTab.GetNumRecords
'aVTab.GetSelection.And(aVTab.GetDefBitmap)' The selection can include some
undefined records!
N = aVTab.GetSelection.Count

```

```

'
' Capture the sort order.
'
av.ClearStatus
av.ShowMsg("Capturing the sort order...")
av.ShowStopButton

lstOrder = {}
for each theRow in 0..(N-1)
  if (av.SetStatus(100*(theRow+1)/N).not) then break end ' 22 Feb 2000

  lstOrder.Add(aTable.ConvertRowToRecord(theRow)) ' Compute this before
  adding a record screws it up!
end
if (lstOrder.Count < N) then ' 22 Feb 2000
  return nERR_USER
end
'
' Find the editable fields (22 Feb 2000: an optimization).
'
lstFields = {}
for each f in aVTab.GetFields
  if (f.IsEditable) then
    lstFields.Add(f)
  end
end
'
' Minimize the table window to avoid redraw delays.
' (22 Feb 2000)
'
aTable.GetWin.Close ' Does not work when aTable was created from a theme in
this script!
'
' Append copies of the selected records to the table, in the correct order.
'
av.ClearStatus
av.ShowMsg("Now editing the table...")
av.ShowStopButton
nCode = nSUCCESS
for each theRow in 0..(N-1)
  if (av.SetStatus(100*(theRow+1)/N).not) then
    nCode = nERR_USER
    break
  end ' 22 Feb 2000

  if (aVTab.CanAddRecord.Not) then
    MsgBox.Error("Unable to add a new record--you may be out of disk space
    or RAM; aborting...",
      sTitle)
    nCode = nERR_ADD
    break
  end
end

theRec = lstOrder.Get(theRow)

aVTab.BeginTransaction
  recDest = aVTab.AddRecord
  for each f in lstFields
    aVTab.SetValue(f, recDest, aVTab.ReturnValue(f, theRec))
  end
aVTab.EndTransaction

end ' Appending selected records
'
' Process any errors that occurred.
'
if (nCode <> nSUCCESS) then ' 22 Feb 2000

```

```

    aVTab.StopEditingWithRecovery(false)
    if (wasAdded) then
        av.GetProject.RemoveDoc(aTable)
    else
        aTable.GetWin.Open
    end
    return nCode
end
'
' Remove the original versions of the records.
'
av.ClearStatus
av.ShowMsg("Cleaning up the table...")
aVTab.BeginTransaction
    aVTab.RemoveRecords(aVTab.GetSelection)
aVTab.EndTransaction
if (aVTab.StopEditingWithRecovery(true).Not) then ' 22 Feb 2000
    msgBox.Error("Unable to save the changes; aborting...", sTitle)
    return nERR_SAVE
end
'
' Update the selection and definition bitmaps.
'
if (wasUnselected.Not) then
    aVTab.GetSelection.SetRange(M-N, N)
    aVTab.UpdateSelection
end
aVTab.SetDefinition(aVTab.GetDefinition)
aVTab.UpdateDefBitmap
'
' Synchronize the disk and in-memory versions of the VTab.
'
aVTab.Flush
'
' Clean up the project if necessary.
'
if (wasAdded) then
    av.GetProject.RemoveDoc(aTable)
else
    aTable.GetWin.Open
end
return nSUCCESS
' end of script

'-----Scriptende-----

' Name : SIS.TableDoc
'
' Zweck: Tabellendokumentation
'
' Programmsequenz aus ArcView-GIS
' /Avenue Programmiers-Reference
' Benötigt das kompilierte Script TableReportFile

' Variablen initialisieren

t = "SIS-Tabelleninformation"

rpt = ""

' Dieses Script aufrufbar oder standalone machen

parm = SELF
if(parm.Is(Table).Not) then
    p = av.GetProject

```

```

docList = p.GetDocs
tblList = List.Make
for each d in docList
  if(d.Is(Table)) then tblList.Add(d) end
end
if(tblList.Count = 0) then
  MsgBox.Warning("In diesem Projekt gibt es keine Tabellen: Abbruch",t)
  exit
end
parm = MsgBox.List(tblList,"Tabelle auswählen:",t)
if(parm = nil) then
  MsgBox.Warning("Keine Tabelle ausgewählt: Abbruch",t)
  exit
end
end

rpt = av.Run("SIS.TableReportFile",parm)
if(SELF.is(Table).Not) then
  MsgBox.Report(rpt,t)
  out = FileDialog.Put("c:tempout.tbl".asFileName,"*.tbl",t)
  if(out = nil) then exit end
  outFile = LineFile.make(out,#FILE_PERM_WRITE)
  if(outFile = nil) then exit end
  outFile.Write(rpt.asTokens(nl),rpt.asTokens(nl).Count)
  outFile.Flush
  outFile.Close
else
  return rpt
end

'-----Scriptende-----

' Name: SIS.TableReportFile
'
' Zweck: Tabellendokumentation
'
' Programmsequenz aus ArcView-GIS
' /Avenue Programmiers-Reference
' Script muss von einem anderen Script aus
' aufgerufen werden, welches den Tabellennamen
' übergibt und das Ergebnis anzeigt
' Variablen Initialisieren

t = "TableReportFile"

rpt = ""

theTable = SELF

' Projekt ermitteln

theProj = av.GetProject

' VTab ermitteln

theVTab = theTable.GetVTab

' Ermitteln der VTab-Felder

theFieldList = theVTab.GetFields
theDict = Dictionary.Make(theFieldList.Count + 1)
theDict.Add ("ALL FIELDS", "ALL FIELDS")
for each i in theFieldList
  theDict.Add(i.GetName,i)
end

```

```

' Dictionary für Enumerations der Feldtypen anlegen

fldDict = Dictionary.Make(19)
fldDict.Add(#FIELD_BYTE,"Ein-Byte Integer")
fldDict.Add(#FIELD_CHAR,"String fixer Länge")
fldDict.Add(#FIELD_DATE,"Datum im Format JJJJMMTT")
fldDict.Add(#FIELD_DECIMAL,"Fließkommazahl im BCD-Format")
fldDict.Add(#FIELD_DOUBLE,"Fließkommazahl Double")
fldDict.Add(#FIELD_FLOAT,"Fließkommazahl Single")
fldDict.Add(#FIELD_ISODATE,"Datum im Stringformat JJJJ-MM-TT")
fldDict.Add(#FIELD_ISODATETIME,"Datum/Uhrzeit im Stringformat
JJJJ-MMM-TT-HH-MM-SS")
fldDict.Add(#FIELD_LOGICAL,"Ein-Byte Boolean")
fldDict.Add(#FIELD_LONG,"Long Integer")
fldDict.Add(#FIELD_MONEY,"Fixkommazahl-String mit 2 Nachkommastellen")
fldDict.Add(#FIELD_SHAPELINE,"Shape/Polyline")
fldDict.Add(#FIELD_SHAPEMULTIPOINT,"Shape/Multi-Point")
fldDict.Add(#FIELD_SHAPEPOINT,"Shape/Point")
fldDict.Add(#FIELD_SHAPEPOLY,"Shape/Polygon")
fldDict.Add(#FIELD_SHORT,"Short Integer")
fldDict.Add(#FIELD_UNSUPPORTED,"Nicht unterstützt")
fldDict.Add(#FIELD_VCHAR,"String variabler Länge")

theFLDList = theDict.ReturnKeys
rpt = rpt + "Tabellename      :" ++ theTable.GetName + nl
rpt = rpt + nl
rpt = rpt + "Ersteller          :" ++ theTable.GetCreator + nl
rpt = rpt + "Erstellungsdatum  :" ++ theTable.GetCreationDate + nl
rpt = rpt + "Kommentar           :" ++ theTable.GetComments + nl
rpt = rpt + "Objekt Tag          :" ++ theTable.GetObjectTag.asstring + nl
rpt = rpt + "DocGUI              :" ++ theTable.GetGui + nl
rpt = rpt + "Tabellenjoin        :" ++ theVTab.IsBase.NOT.asstring + nl
rpt = rpt + "Tabellenlink        :" ++ theVTab.IsLinked.asstring + nl
rpt = rpt + "Anzahl der Felder:" ++ theFieldList.Count.Asstring + nl
rpt = rpt + nl

' Der Report aller Felder in der VTab

for each i in theFieldList
  theField = i
  rpt = rpt + "FELDNAME      :" ++ theField.GetName + nl
  rpt = rpt +
  "-----" + nl
  rpt = rpt + "Datentyp      :" ++ fldDict.Get(theField.GetType) + nl
  rpt = rpt + "Breite        :" ++ theField.GetWidth.AsString + nl
  rpt = rpt + "Präzision     :" ++ theField.GetPrecision.AsString + nl
  rpt = rpt + "Alias         :" ++ theField.GetAlias + nl
  rpt = rpt + "Sichtbar     :" ++ theField.IsVisible.AsString + nl
  rpt = rpt + "Pixelbreite  :" ++ theField.GetPixelWidth.AsString + nl
  rpt = rpt + nl
end
rpt = rpt + nl
return rpt

'----Scriptende----

' Name:  SIS.TrnTblThemeMake
'
' Zweck: Erstellen eines Punktthemas für die Definition von Abbiegeregeln
'        für den Straßengraphen
'
'
'
' Thema: SIS, Netzwerk
'

```

```

' Voraussetzung: SIS.MkFTheme wird aufgerufen
'
'
' Self:
'
' Rückgabe:
'
' Autor: Horst Baumgartner 2001/11/21
'

theFieldList = {
Field.Make("ID",#FIELD_DOUBLE,12,0),
Field.Make("F_EDGE",#FIELD_DOUBLE,12,0),
Field.Make("T_EDGE",#FIELD_DOUBLE,12,0),
Field.Make("SECONDS",#FIELD_DOUBLE,12,0),
Field.Make("User",#FIELD_CHAR,10,0),
Field.Make("Date",#FIELD_ISODATE,10,0)
}

av.Run("SIS.MkFTheme",{ "Straßengraphen
anlegen","strassen.shp",theFieldList, Point})

'-----Scriptende-----

' Name: SIS.ViewDP.List
'
' Zweck: Sucht in einem View alle verfügbaren Punktthemen
'        aus einer dynamischen Segmentierung
'
'
' Thema: SIS, Dynamische Segmentierung
'
' Voraussetzung: Das aktive Dokument muss ein View sein.
'
' Self:
'
' Rückgabe: Liste der Themennamen
'
' Autor: Horst Baumgartner 2000/02/22
'

theThmList = List.Make

theView = av.GetActiveDoc

for each thm in theView.GetThemes
  theSrcName = thm.GetSrcName
  if(theSrcName.Is(DynName)) then
    if(theSrcName.IsPoint) then
      theThmList.Add(thm.AsString)
    end
  end
end

return(theThmList)
'SISExt.CanUnload

return(System.CanUnloadLibrary(self.GetDependencies.get(0)))

'-----Scriptende-----

' Name: SISExt.Install

```

```

' Zweck: Installiert die Elemente,
'       wenn ein Projekt geöffnet ist

if(av.GetProject = nil) then
    return nil
end

' Elemente holen und in der Oberfläche einsetzen

ViewGUI = av.GetProject.FindGUI("View")
theMenuBar = ViewGUI.GetMenuBar
theMenu = theMenuBar.FindByLabel("Window")
if(theMenu <> nil) then
    theMenuBar.Add(self.Get(12), theMenuBar.getControls.Find(theMenu))
else
    theMenuBar.Add(self.get(12),999)
end

TableGUI = av.getProject.FindGUI("Table")
theTMenuBar = TableGUI.GetMenuBar
theTMenuBar.Add(self.get(13),999)

theBBar = ViewGUI.GetButtonBar
theBBar.Add(self.Get(2), 995)
theBBar.Add(self.get(3), 996)
theBBar.Add(self.get(4), 997)
theBBar.Add(self.get(5), 998)
theBBar.Add(self.get(6), 999)

theTBar = ViewGUI.GetToolBar
theTBar.Add(self.get(7), 995)
theTBar.Add(self.get(8), 996)
theTBar.Add(self.get(9), 997)
theTBar.Add(self.get(10), 998)
theTBar.Add(self.get(11), 999)

' Benutzererkennung beim Laden festlegen
' und in der globalen Variable _theUser speichern

av.run("SIS.SetUser",{ })

'----Scriptende-----

' Name: SISExt.Make
' Zweck: Erstellt das Extension-Objekt für SIS

SISExt = Extension.Make(
    "$AVHOME/ext32/SIS.avx".AsFileName,
    "SIS",
    av.FindScript("SISExt.Install"),
    av.FindScript("SISExt.Uninstall"),
    {"$AVBIN/avdlog.dll".AsFileName})

SISExt.SetExtVersion(1)
SISExt.SetAbout("SIS-Straßeninformations-System Steyr (v1.41)")
SISExt.SetUnloadScript(av.FindScript("SISExt.Unload"))
SISExt.SetCanUnLoadScript(av.FindScript("SISExt.CanUnload"))

' Straßenverzeichnis wird geleert

aDialog = av.FindDialog("SISKt")
aDialog.FindByName("lbox_StrVerz").Empty
SISExt.Add(aDialog) ' 0

bDialog = av.FindDialog("SISBP")
SISExt.Add(bDialog) ' 1

```

```

' Buttons im View-Menü anhängen

theBBar = av.getProject.FindGUI("View").GetButtonBar
SISExt.Add(theBBar.FindByScript("SIS.CalcNodes"))           ' 2
SISExt.Add(theBBar.FindByScript("SIS.Kt.Recalc"))           ' 3
SISExt.Add(theBBar.FindByScript("SIS.MeasurePolyLine"))     ' 4
SISExt.Add(theBBar.FindByScript("SIS.RtDef"))               ' 5
SISExt.Add(theBBar.FindByScript("SIS.KtUnion"))             ' 6

' Tools aus dem View Menü holen und hinzufügen

theTBar = av.getProject.FindGUI("View").GetToolBar
SISExt.Add(theTBar.FindByScript("SIS.BPSet"))               ' 7
SISExt.Add(theTBar.FindByScript("SIS.RtRemove"))            ' 8
SISExt.Add(theTBar.FindByScript("SIS.Kt.Set"))              ' 9
SISExt.Add(theTBar.FindByScript("SIS.Split"))               ' 10
SISExt.Add(theTBar.FindByScript("SIS.StrInvalidByPoint"))   ' 11

' SIS Menüleiste anhängen
theMenuBar = av.getProject.FindGui("View").getMenuBar
SISExt.Add(theMenuBar.FindByLabel("SIS"))                   ' 12

' Tabellen Tool-Menü

theTMenuBar = av.getProject.FindGui("Table").getMenuBar     ' 13
SISExt.Add(theTMenuBar.FindByLabel("Tools"))

' Alle Scripts mit dem Prefix "SIS" abfragen

for each aDoc in av.getProject.GetDocs
  if(aDoc.Is(SEd)) then
    if(aDoc.GetName.Contains("SIS.")) then
      if(aDoc.IsCompiled.Not) then
        aDoc.Compile
      end
      SISExt.Add(av.FindScript(aDoc.GetName))
    end
  end
end

SISExt.Commit

'-----Scriptende-----

'Name: SISExt.UnInstall
'
'
if(av.getProject = nil) then
  return nil
end

if(av.getProject.IsClosing) then
  return nil
end

' Die Teile wieder entfernen

ViewGUI = av.getProject.FindGUI("View")

theMenuBar = ViewGUI.GetMenuBar
theMenuBar.Remove(self.Get(12))

```

```

TableGUI = av.GetProject.FindGUI("Table")

theTMenuBar = TableGUI.GetMenuBar
theTMenuBar.Remove(self.Get(13))

theBBar = ViewGUI.GetButtonBar
theBBar.Remove(self.Get(2))
theBBar.Remove(self.get(3))
theBBar.Remove(self.get(4))
theBBar.Remove(self.get(5))
theBBar.Remove(self.get(6))

theTBar = ViewGUI.GetToolBar
theTBar.Remove(self.get(7))
theTBar.Remove(self.get(8))
theTBar.Remove(self.get(9))
theTBar.Remove(self.get(10))
theTBar.Remove(self.get(11))

'----Scriptende-----

'Name: SISExt.Unload
'
Dialog.DetachFromExtension(self)
' Globale Variable für die Benutzererkennung entfernen
_theUser = nil
'----Scriptende-----

' Name: Segment2Point
'
' Zweck: Erzeugen einer Punktliste aus einer sortierten Shapedatei vom Typ
Polyline
'
'
' Thema: SIS, Sonstiges
'
' Voraussetzung: Das aktive Dokument muss ein View sein.
'                 Das aktive Thema muss vom Typ Polyline sein
'
' Self:
'
' Rückgabe:
'
' Autor: Horst Baumgartner 2002/01/03
'
'

theView = av.getActiveDoc
theTheme = theView.GetActiveThemes.Get(0)
theFtab = theTheme.getFTab

LinFld = theFtab.FindField("Info")
ShpFld = theFtab.FindField("Shape")

```

```

myFile = FileDialog.Put("points.dbf".AsFileName,"*.*","File")
theVTab = VTab.MakeNew(myFile,dbase)
myTable = Table.make(theVTab)

NlinFld = Field.Make("ID",#FIELD_CHAR,20,0)
XFld = Field.Make("X",#FIELD_DOUBLE,16,2)
YFld = Field.Make("Y",#FIELD_DOUBLE,16,2)
OrdFld = Field.Make("Order",#FIELD_DOUBLE,12,0)

theVTab.AddFields({NlinFld,XFld,YFld,OrdFld})

LastLine = "Dummy"
theOrder = 1

for each rec in theFtab
  theShape = theFtab.ReturnValue(ShpFld,rec)
  thePList = theShape.AsList
  theLine = theFtab.ReturnValue(LinFld,rec)
  for each ln in thePList
    if(LastLine <> theLine) then
      LastLine = theLine
      theOrder = 1
    else
      ln.Remove(0)
    end
    for each pt in ln
      nrec = theVTab.AddRecord
      theVtab.SetValue(NlinFld,nrec,theLine)
      theVtab.SetValue(XFld,nrec,pt.GetX)
      theVtab.SetValue(YFld,nrec,pt.GetY)
      theVtab.SetValue(OrdFld,nrec,theOrder)
      theOrder = theOrder+1
    end
  end
end
end

'-----Scriptende-----

' Name:   SIS.RtToPolyline
'
' Zweck: Wandelt ein Routenthema in ein Shapefile vom Typ Polyline
'        um
'
'
' Thema:  SIS, Dynamische Sementierung
'
' Voraussetzung: Das aktive Dokument muss ein View sein. Die Elemente der
'                Segementtabelle müssen geordnet sein (Feld [Order])
'                SIS.ViewDPln.List und
'                SIS.MkFTheme werden aufgerufen
'
' Self:
'
' Rückgabe: keine
'
' Autor:  Horst Baumgartner 2002/01/20
'

t="SIS-Route zu Polyline"

theView = av.GetActiveDoc

theThemeString =
MsgBox.ListAsString(av.run("SIS.ViewDPln.List",{ }),"Routenthema
auswählen",t)

```

```

' Benutzerabbruch

if(theThemeString = nil) then
  exit
end

theTheme = theView.FindTheme(theThemeString)

' Ftab holen

theFtab = theTheme.getFtab
theShapeFld = theFtab.FindField("Shape")

' Index für schnellere Suche aufbauen

theFRtIdFld = theFtab.FindField("Rid")
if(theFtab.IsFieldIndexed(theFRtIdFld).Not) then
  theFtab.CreateIndex(theFRtIdFld)
end

' zugehörige Routentabelle auswählen

theRtTableString =
MsgBox.ListAsString(av.Run("SIS.Table.List",0),"Zugehörige Routentabelle
auswählen:",t)

' Benutzerabbruch

if(theRtTableString = nil) then
  exit
end

theRtTable = av.GetProject.FindDoc(theRtTableString)
theRtVtab = theRtTable.GetVtab

' Felder für die Routentabelle

RidFld = Field.Make("Rid",#FIELD_DOUBLE,16,0)
BPFfromFld = Field.Make("BPFfrom",#FIELD_DOUBLE,16,0)
BPTofFld = Field.Make("BPTof",#FIELD_DOUBLE,16,0)
RlenFld = Field.Make("Rlen",#FIELD_DOUBLE,16,2)
RInfoFld =Field.Make("Info",#FIELD_CHAR,30,0)
RUserFld =Field.Make("User",#FIELD_CHAR,10,0)
RDateFld =Field.Make("Date",#FIELD_DATE,8,0)

theFieldList =
{RidFld,BPFfromFld,BPTofFld,RlenFld,RInfoFld,RUserFld,RDateFld}

' Thema vom Typ Polyline anlegen

thePFtab = av.Run("SIS.MkFTheme",{ "Neues Linienthema", "rtpline.shp",
  theFieldList, Polyline})

' Alle Segmente einer Route zu einer Polylinie zusammenfassen

theBitmap = theFtab.getSelection

for each rec in theRtVtab
  theRid = theRtVtab.ReturnValue(theRtVtab.FindField("Rid"),rec)
  theExpr = "([Rid] = "+ theRid.AsString + ")"
  theFtab.Query(theExpr,theBitmap, #VTAB_SELTYPE_NEW)

' Liste leeren

```

```

thePtList = {}

' Segmentthema durchlaufen

isStart = true

frec = -1

isStart = true

while(true)
  frec = theBitmap.getNextSet(frec)
  if(frec = -1) then
    break
  else
    ' Punkteliste aufbauen
    theShape = theFtab.ReturnValue(theShapeFld,frec).AsPolyline

    if(theFtab.ReturnValue(theFtab.FindField("Direction"),frec) = -1)
    then
      theShape = theShape.Flip
    end

    for each ln in theShape.AsList
      if(isStart = false) then
        ln.Remove(0)
      else
        isStart = false
      end
      thePtList = thePtList + ln
    end
  end
end
nrec = thePftab.AddRecord

thePftab.SetValue(thePftab.FindField("Shape"),nrec,PolyLine.Make({thePtList}))
for each fld in theFieldList

  thePftab.SetValue(thePftab.FindField(fld.GetAlias),nrec,theRtVtab.ReturnValue(theRtVtab.FindField(fld.GetAlias),rec))
end
end
theFtab.Flush
theView.getEditableTheme.StopEditing(true)
theView.SetEditableTheme(NIL)

'-----Scriptende-----

' Name:  SIS.ViewDPln.List
'
' Zweck: Sucht in einem View alle verfügbaren Linienthemennamen
'        aus einer dynamischen Segmentierung
'
'
' Thema: SIS, Dynamische Segmentierung
'
' Voraussetzung: Das aktive Dokument muss ein View sein.
'
' Self:
'
' Rückgabe: Liste der Themennamen
'
' Autor: Horst Baumgartner 2002/01/20
'

theThmList = List.Make

```

```
theView = av.GetActiveDoc

for each thm in theView.GetThemes
  theSrcName = thm.GetSrcName
  if(theSrcName.Is(DynName)) then
    if(theSrcName.IsLinear) then
      theThmList.Add(thm.AsString)
    end
  end
end

return(theThmList)

'-----Scriptende-----'
```