

Effektivisering vid behandling av CONTRAM-data i ArcMap

Examensarbete inom civilingenjörsprogrammet Väg- och vattenbyggnad

ANNA PERSSON OCH MARIA SANDÉN

Institutionen för bygg- och miljöteknik
Avdelningen för geologi och geoteknik
Väg och trafik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg 2007
Examensarbete 2007:131

EXAMENSARBETE 2007:131

Effektivisering vid behandling av CONTRAM-data i ArcMap

Examensarbete inom civilingenjörsprogrammet Väg- och vattenbyggnad

ANNA PERSSON OCH MARIA SANDÉN

Institutionen för bygg- och miljöteknik
Avdelningen för geologi och geoteknik
Väg och trafik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, 2007

Effektivisering vid behandling av CONTRAM-data i ArcMap

Examensarbete inom civilingenjörsprogrammet Väg- och vattenbyggnad
ANNA PERSSON OCH MARIA SANDÉN

© ANNA PERSSON OCH MARIA SANDÉN, 2007

Examensarbete 2007:131
Institutionen för bygg och miljöteknik
Avdelningen för geologi och geoteknik
Väg och trafik
Chalmers tekniska högskola
412 96 Göteborg
Telefon: 031-772 10 00

Omslag: CONTRAM-nät över Linnégatan i Växjö, Trafik, Ramböll Sverige AB,
Malmö.

Chalmers Reproservice
Göteborg 2007

Efficiency improvement concerning transferring CONTRAM-data to ArcMap

Master's Thesis in Civil Engineering
ANNA PERSSON OCH MARIA SANDÉN
Department of Civil and Environmental Engineering
Division of GeoEngineering
Road and Traffic Group
Chalmers University of Technology

ABSTRACT

Traffic planning is constantly developing in order to become more efficient. This master's thesis creates a tool which aims to improve traffic planning, focusing on the connection between the traffic simulation program CONTRAM and the GIS program ArcMap.

This master's thesis describes those programs needed in order to export data from CONTRAM to ArcMap, and how the data export, via the transformation program Con2arcView, takes place today. Data from CONTRAM is first exported to ArcMap through Con2arcView, and then to Excel to be worked on. Last, the changed data is brought back to ArcMap.

In order to make the data transfer from CONTRAM to ArcMap more efficient, a number of commands have been created through Visual Basic for applications. Because of the new commands the calculation earlier carried out in Excel is no longer needed. Through the commands the user may receive information about the total number vehicles on any road by the click of a button.

The data transfer from CONTRAM to ArcMap has become more efficient through this master's thesis since the number of steps in order to export data from CONTRAM to ArcMap have been reduced. As a result of this, traffic planning has become easier and more efficient. The risk of errors occurring has been reduced as the user only has to carry out a few steps in order to transfer data from CONTRAM to ArcMap.

Key words: Geographic information system, GIS, traffic planning, Visual Basic for applications, CONTRAM, ArcMap, development of the connection between CONTRAM and ArcMap

Effektivisering vid behandling av CONTRAM-data i ArcMap

Examensarbete inom civilingenjörsprogrammet Väg- och vattenbyggnad
ANNA PERSSON OCH MARIA SANDÉN
Institutionen för bygg- och miljöteknik
Avdelningen för geologi och geoteknik
Väg och trafik
Chalmers Tekniska Högskola

SAMMANFATTNING

Trafikplaneringsarbetet utvecklas ständigt för att bli mer effektivt. Examensarbetet syftar till att försöka effektivisera trafikplaneringsarbetet, och då främst kopplingen mellan trafikanalysovertyget CONTRAM och GIS-programvaran ArcMap.

I examensarbetet beskrivs de programvaror som behövs för att exportera data från CONTRAM till ArcMap, samt hur exporteringen från CONTRAM via transformationsprogrammet Con2arcView till ArcMap går till idag. Data från CONTRAM exporteras med hjälp av Con2arcView till ArcMap och exporteras därefter till Excel för bearbetning, för att sedan återföras till ArcMap.

För att effektivisera överföringen av data från CONTRAM till ArcMap har ett antal kommandon skapats genom programmeringsspråket Visual Basic for applications som finns inbyggt i ArcMap. Genom de olika kommandona elimineras moment som tidigare utfördes i Excel. Med de nya kommandona kan användaren, direkt efter att data exporterats till ArcMap, klicka på en knapp för att få de totala flödena på alla gator.

Genom examensarbetet har kopplingen mellan CONTRAM och ArcMap effektiviserats genom att antalet moment som tidigare krävdes för att exportera data från CONTRAM till ArcMap har minskats. Detta har gjort att trafikplaneringsarbetet har blivit enklare och mer effektivt. Risken för fel har även minskat då användaren genom de nya kommandona endast behöver utföra ett fåtal arbetsmoment för att data ska kunna överföras från CONTRAM till ArcMap.

Nyckelord: Geografiska informationssystem, GIS, trafikplanering, Visual Basic for applications, CONTRAM, ArcMap, utveckling av koppling mellan CONTRAM och ArcMap

Innehåll

ABSTRACT	I
SAMMANFATTNING	II
INNEHÅLL	III
FÖRORD	V
BEGREPPSFÖRKLARING	VII
1 INLEDNING	1
1.1 Bakgrund	1
1.2 Syfte	1
1.3 Avgränsningar	2
1.4 Tillvägagångssätt	2
1.5 Aktuell data	2
1.6 Disposition	2
2 GEOGRAFISKA INFORMATIONSSYSTEM	4
2.1 Beståndsdelar	4
2.2 Användningsområden	5
2.3 Modeller med GIS	5
3 ARCGIS	7
3.1 Desktop-GIS	7
3.1.1 Applikationer till Desktop-GIS	7
4 CONTRAM	9
4.1 Noder	9
4.2 Länkar	10
4.3 Zoner	11
4.4 Tidsintervall	11
5 CON2ARCVIEW	13
5.1 Överföring av data från CONTRAM till ArcView idag	14
5.1.1 Beräkningsblad	14
6 VISUAL BASIC	16
6.1 Visual Basic for Applications	16
6.2 Visual Basic Editor	16

6.2.1	Projekt	17
6.2.2	Formulär	17
6.2.3	Procedur	18
6.2.4	Modul	18
6.2.5	Användarkontroller	19
6.3	ArcObjects	20
6.3.1	Feature class	20
6.3.2	Feature	20
6.3.3	Cursor	21
6.3.4	QueryFilter	21
6.3.5	Calculator	21
6.3.6	Fields	22
7	UTFÖRD PROGRAMMERING I VBA OCH ARCMAP	23
7.1	Attributtabellen	23
7.1.1	Kolumner hos attributtabellen för NOR10Arrivals	23
7.1.2	Skapade kolumner i attributtabellen för NOR10Arrivals	24
7.2	Beräkning av det totala flödet hos varje enskild länk	24
7.3	Skapandet av Noder 1 och Noder 2	24
7.4	Beräkning av det totala flödet mellan ett nodpar	25
7.5	Öppna attributtabellen	26
7.6	Radera fält	26
8	RESULTAT OCH SLUTSATSER	28
8.1	Resultat	28
8.2	Slutsatser	28
9	FÖRSLAG TILL FORTSATT ARBETE	30
9.1	Skapa nytt lager	30
9.2	Skapa nya punkter	30
9.3	Skapa nya linjer	31
9.4	Skapa ny tabell	31
9.5	Utveckling av länknätverk och attributtabell	31
9.6	Förslag till användningsområden för ”Nya linjer” och ”Nya punkter”	32
9.7	Övriga förbättringar	32
10	REFERENSER	33
11	BILAGOR	36

Förord

Detta examensarbete utgör den avslutande delen av vår civilingenjörsutbildning på Väg- och Vattenbyggnadsprogrammet vid Chalmers tekniska högskola i Göteborg.

Arbete är gjort på uppdrag av Trafik, Ramböll Sverige AB i Malmö, och är utfört vid Mät- och kart, Ramböll Sverige AB i Göteborg, sommaren och hösten 2007.

Ett stort tack riktas till följande personer som hjälpt oss under arbetets gång:

Sven-Olav Johansson, enhetschef Mät- och kart, Ramböll Sverige AB i Göteborg

Hamid Rezaie, enhetschef Trafik, Ramböll Sverige AB i Malmö

Anna Windal, trafikplanerare, Ramböll Sverige AB i Malmö

Gunnar Lannér, universitetslektor, Chalmers tekniska högskola

Göteborg, december 2007

Anna Persson och Maria Sandén

Begreppsförklaring

GIS	Geografiskt informationssystem, är ett datorbaserat informationssystem för att samla in, lagra, bearbeta, analysera och presentera lägesbunden information
ArcMap	En GIS-applikation som används för analys och redigering av kartor samt geografiska data
CONTRAM	Trafikanalysverktyg som kan användas för att analysera olika trafikmiljöer
Con2arcView	Transformationsprogram som exporterar data från CONTRAM till ArcMap
VBA	Visual Basic for applications, är ett objektorienterat programmeringsspråk
Nod	Representerar en korsning i ett CONTRAM-nätverk
Länk	Används i CONTRAM för att representera körfält
Zon	En zon är en punkt som antingen genererar eller attraherar trafik i ett CONTRAM-nät

1 Inledning

1.1 Bakgrund

Geografiska informationssystem, hädanefter benämnt GIS, har använts i Sverige sedan mitten av 1980-talet,¹ och sedan dess har användningen av GIS, och dess programvaror ökat, främst under de senaste åren. Programvarorna används inom många olika områden och är utvecklade för att tillgodose en bred marknad. Inom trafikplaneringsområdet används GIS-programvaror i stor utsträckning för olika typer av uppgifter såsom flödes- och tillgänglighetsstudier. Kraven på GIS-programvaror ökar ständigt då utveckling inom trafikplaneringsområdet hela tiden sker. Programmen skall vara enkla att lära sig och vara användarvänliga.

Trafikanalyser av trafikförändringar och flöden i städer kan utföras i programmet CONTRAM, Continuous Traffic Assignment Model. Analyser i CONTRAM resulterar i ett nätverk uppbyggt av länkar och noder med tillhörande attribut. Noderna representerar korsningar och länkarna vägar eller gator i nätverket. Attribut kan vara hastighet eller flöde hos en specifik länk, eller vilken korsningstyp en specifik korsning har.

Trafikanalyser sker idag bland annat i CONTRAM, medan det slutgiltiga resultatet presenteras i en GIS-programvara, exempelvis programmet ArcMap. CONTRAM är till stor del uppbyggt för analyser av trafik, och målgruppen för användandet av programmet är trafikplanerare och ingenjörer. Detta medför att programmets resultat är svåröverskådliga för de som ska fatta beslut, ofta politiker, med just dessa resultat som grund.

För att kunna arbeta med data som genereras från analyser i CONTRAM, i ArcMap, har transformationsprogrammet Con2arcView skapats. Programmet exporterar informationen från CONTRAM som en databasfil till ArcMap så att data kan bearbetas i programmet. För att kunna bearbeta informationen i ArcMap måste informationen först redigeras i ett flertal olika moment i Excel, för att därefter återigen exporteras till ArcMap. Ett beräkningsblad är därför skapat som gör de beräkningar som krävs för att data av intresse från CONTRAM skall kunna användas i ArcMap. Efter att data bearbetats i beräkningsbladet exporteras den till ArcMap och vidare arbete i ArcMap kan påbörjas.

1.2 Syfte

Syftet med examensarbetet är att förenkla och underlätta processen vid export av för användaren relevant information från CONTRAM till ArcMap genom att minska antalet moment för att exportera med hjälp av nya menyknappar som utför flera steg.

¹ Future position Xs Hemsida, 2007-06-12

1.3 Avgränsningar

Examensarbetet har tagit hänsyn till följande avgränsningar:

- I examensarbetet används ArcMap som GIS-programvara, och jämförelser med andra programvaror görs inte.
- Examensarbetet avgränsas till att endast studera hur data från trafikanalysverktyget CONTRAM, med hjälp av transformationsprogrammet Con2arcView, exporteras till ArcMap. Jämförelser med data från andra trafikanalysprogram görs ej.
- Programmeringsspråket som används i ArcMap är Visual Basic for Applications, VBA.
- Vid programmeringen har det endast tagits hänsyn till ett specifikt nätverk, och vid andra indata krävs modifiering av VBA-koden.

1.4 Tillvägagångssätt

Arbetet inleddes med en litteraturstudie som sedan pågått parallellt med det övriga arbetet under hela projektets gång. Litteratur har sökts på Internet och i biblioteksdataserna Libris och CHANS. De dokument som studerats har bestått i rapporter och litteratur som är relevant för ämnet. Information har även sökts på programvarornas egna hemsidor. Hjälp har sökts via communities och forum på Internet. Efter litteraturstudien fortsatte arbetet med en programmeringsfas, där kommandon utvecklades och som programmerades i ArcMap med programmeringsspråket VBA. Arbetet avslutades med en resultat- och slutsatsstudie.

1.5 Aktuell data

För att kunna utföra arbetet krävs ett nätverk uppbyggt i CONTRAM. Under hela arbetets gång har ett CONTRAM-nät över Linnégatan i Växjö funnits tillgängligt. Nätet är från trafikplaneringsavdelningen vid Ramböll Sverige i Malmö. Därifrån har även beräkningbladet i Excel och data som krävs för att kunna utföra programmeringen i ArcMap erhållits, såsom attributtabell och karta över Linnégatan i Växjö.

1.6 Disposition

Efter det inledande kapitlet beskrivs övergripande i kapitel två hur geografiska informationssystem används och vilka användningsområden som finns. Även beståndsdelarna i ett GIS beskrivs kortfattat.

I kapitel tre beskrivs GIS-programvaran ArcGIS med tillhörande produkter och applikationer. Funktioner och användningsområden för CONTRAM redogörs för i kapitel fyra. I kapitel fem beskrivs överföringsprogrammet Con2arcView. Även hur exporteringen av data från CONTRAM till ArcMap fungerar i dagsläget förklaras. Beräkningsbladet som används vid exporten av data beskrivs. Programmeringsspråket Visual Basics och VBA beskrivs i kapitel 6. Uppbyggnad och de kommandon som använts i examensarbetet beskrivs.

I kapitel sju beskrivs de nya programdelar som skapats. Kapitel åtta är en resultat- och slutsatsdel som diskuterar de resultat som ges ur examensarbetet. Vilka slutsatser som kan dras genom examensarbetet tas även upp. Det avslutande kapitlet behandlar hur vidare arbete skulle kunna bedrivas

2 Geografiska informationssystem

Geografiska informationssystem är ett datorbaserat informationssystem för att samla in, lagra, bearbeta, analysera och presentera lägesbunden information². Med lägesbunden information avses allt som kan knytas till en specifik position eller plats, till exempel var en skola är belägen eller hur stor area ett sjöområde har. Ofta presenteras resultaten från GIS med hjälp av kartor.³

Geografiska informationssystem kan användas vid analys, beräkning och analys av olika typer av problem. GIS kan även användas som hjälp vid olika typer av beslut som skall fattas. Med hjälp av GIS förenklas hanteringen av information, genom att informationen blir mer lättillgänglig för många användare.

En stor skillnad mellan presentation av information på vanliga kartor och presentation med hjälp av GIS-program är att information som visas på en vanlig karta inte kan väljas, utan all information på kartan presenteras. Vanliga kartor skall ofta passa en bred grupp av användare, och blir aldrig optimal för någon. Med presentation av informationen i GIS kan det som är av intresse vid olika tillfällen, eller för olika användare, presenteras. Detta kan göras på grund av att inom GIS skiljs lagring och presentation av data åt, vilket gör att den information som är av intresse enkelt kan väljas ut och presenteras.

2.1 Beståndsdelar

För att ett GIS skall fungera krävs att följande beståndsdelar finns,⁴ se även Figur 2.1.

Användaren är den som nyttjar och använder det geografiska informationssystemet.

Mjukvaran utgör redskapet i ett GIS. Mjukvaran skall klara att utföra de operationer och analyser som krävs.

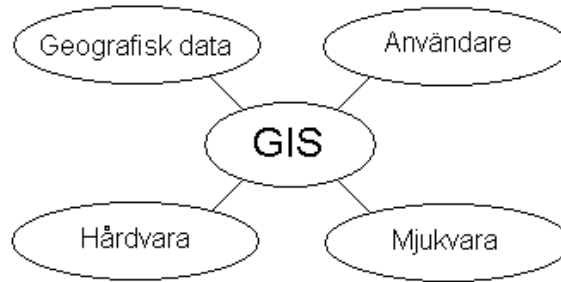
Hårdvara är den dator eller redskap som gör att mjukvaran fungerar. Kravet på hårdvaran är beroende av vilken typ av uppgift som skall utföras.

Geografisk data är den information som skall behandlas.

² Kungliga tekniska högskolans hemsida, 2007-11-26

³ Chalmers tekniska högskolas hemsida, 2007-11-25

⁴ Kungliga tekniska högskolans hemsida, 2007-11-26



Figur 2.1 Beståndsdelar i ett GIS.

2.2 Användningsområden

GIS har funnits i Sverige sedan mitten av 1980-talet och var till en början inte tillgängligt för alla på grund av avancerade och dyra system samt att programmen krävde hög teknisk kunskap. Datorerna har blivit billigare och programvaror mer lättanvända vilket har medfört allt fler användare de senaste åren.⁵

Det första system av GIS som skapades var ett system för landskapsplanering i Kanada på 1960-talet.⁶ I Sverige har GIS först och främst använts för att förenkla framställningen av kartor, dokumentation och visualisering. Idag används GIS i stor utsträckning även som ett presentationsverktyg.

Användningen av GIS sker inom många olika områden där det geografiska läget har betydelse, bland annat hanteras geologiska data och satellitdata ofta härigenom. GIS används inom områden som trafik- och samhällsplanering, kartframställning, visualisering av geografiska data, marknadsföring, försvar, industri, transport och logistik, jordbruk samt räddningstjänst.⁷

2.3 Modeller med GIS

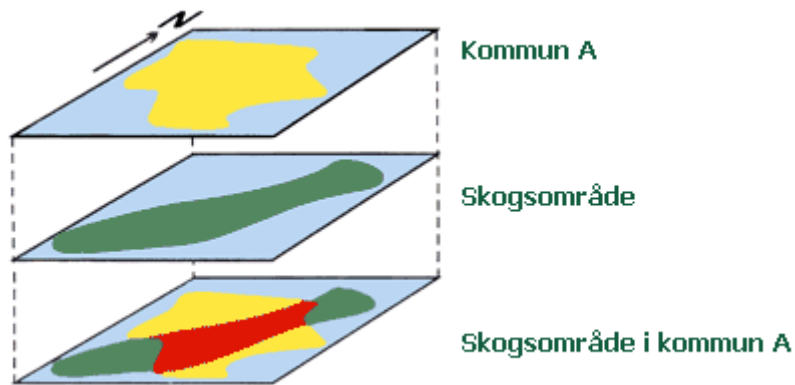
Med hjälp av GIS kan en modell och beskrivning av verkligheten göras. Det är i princip omöjligt att göra en exakt modell av verkligheten och därför krävs viss förenkling och generalisering. Verkligheten delas upp i olika tematiska skikt, där varje tema representerar ett ämne, såsom städer, vägar och befolkning. Varje tema representeras av ett eller flera skikt, där skikten innehåller information om temat. Exempelvis kan temat byggnader delas in i olika skikt, ett för bostäder, ett för skolor och ett för butikslokaler. Fler skikt kan läggas in om modellen kräver det. Genom att

⁵ Future Position Xs hemsida, 2007-06-12

⁶ ESRI:s hemsida, 2007-11-26

⁷ Eklundh 2003, s. 14

flera skikt läggs ovanpå varandra på en bakgrundskarta kan olika typer av analyser och presentationer göras,⁸ se även Figur 2.2 nedan.



Figur 2.2 Uppdelning av verkligheten i tematiska skikt.⁹

⁸ Eklundh 2003, s. 17

⁹ GIS-centrum, Lunds universitets hemsida, 2007-11-26

3 ArcGIS

ArcGIS är en produkt från det amerikanska företaget ESRI, Environmental Systems Research Institute. ArcGIS är ett integrerat system bestående av fyra programvaror som tillsammans bildar ett geografiskt informationssystem.¹⁰ I examensarbetet behandlas endast programvaran Desktop-GIS som bland annat innehåller applikationen ArcMap, vilken har en central roll under arbetets gång.

3.1 Desktop-GIS

Desktop-GIS är ett datorbaserat karthanteringssystem, som är ett verktyg för att hantera, skapa, importera, visualisera, analysera och publicera geografiska data. Desktop-GIS representerar verkligheten på datorn på liknande sätt som vanliga kartor gör, förutom att Desktop-GIS har fördelen att vara mer flexibelt vid redigeringsarbete, eftersom användaren själv kan välja vilken information som skall visas på skärmen.¹¹ I produkten Desktop-GIS ingår följande fyra programvaror:

ArcReader är en gratis programvara för att skriva ut, visa och utforska kartor. I ArcReader är kartorna skrivskyddade.

I **ArcView** har användare möjlighet att skapa och visualisera egna kartor genom att koppla samman olika geografiska data.

ArcEditor är ett verktyg för att bl.a. redigera och hantera geografiska objekt samt skapa geografiska databaser.

ArcInfo är ett komplett system för att skapa och analysera GIS-data. Utöver de funktioner som ArcEditor har, finns även funktioner för avancerad geobearbetning.

3.1.1 Applikationer till Desktop-GIS

ArcView, ArcEditor och ArcInfo består av applikationerna ArcMap, ArcCatalog, ArcToolbox, ModelBuilder och ArcGlobe. Med hjälp av dessa applikationer kan användaren utföra en rad olika operationer t.ex. karthantering, geografiska analyser och geobearbetning.¹² Nedan ges en kort beskrivning av applikationerna.

¹⁰ ESRI's hemsida, 2007-06-15

¹¹ ESRI's hemsida, 2007-06-15

¹² ESRI's hemsida, 2007-06-18

ArcMap är en applikation som används för analys och redigering av kartor samt geografiska data.

ArcCatalog används för att organisera och hantera all GIS-data.

ArcToolbox innehåller geobearbetningsverktyg för bland annat datakonvertering och vektoranalys.

ModelBuilder är ett verktyg för att grafiskt modellera och designa geobearbetningsmodeller.

ArcGlobe är en applikation som gör det möjligt att hantera och visualisera geografisk data i 3D.

4 CONTRAM

CONTRAM, CONTinuous TRaffic Assignment Model, är ett dynamiskt trafikanalysverktyg utvecklat av TRL, Transport Research Laboratory, i Storbritannien.¹³ Programmet används för att analysera trafikflöden, köer, fördröjningar i korsningar, kostnader, bränsleförbrukning och ruttval i mindre trafiknät, som en mindre stad eller en stadsdel i en större stad.¹⁴

Programmet kan modellera ökningen och avtagandet av trängsel som kan uppstå vid rusningstrafik under delar av dagen. Olika korsningstyper så som oregrerade korsningar, cirkulationsplats, korsning med högerregeln och trafiksignalreglerade korsningar kan modelleras. Även olika tidsintervall kan väljas för modelleringen beroende på dygnets olika trafikförhållanden.¹⁵

I CONTRAM finns en tillämpning som gör att användare kan analysera effekterna till följd av en incident. Kölängder eller olyckor, och deras varaktigheter kan leda till att resenärer väljer en annan väg än den normala färdvägen.

För att kunna analysera trafiksituationen för ett önskat område i CONTRAM krävs ett nätverk, som huvudsakligen byggs upp av länkar, noder och zoner som skall representera verkligheten. För att underlätta uppbyggnaden av nätverket och få det att efterlikna verkligheten kan en kartbild över området användas som bakgrundsbild när nätverket byggs upp.

4.1 Noder

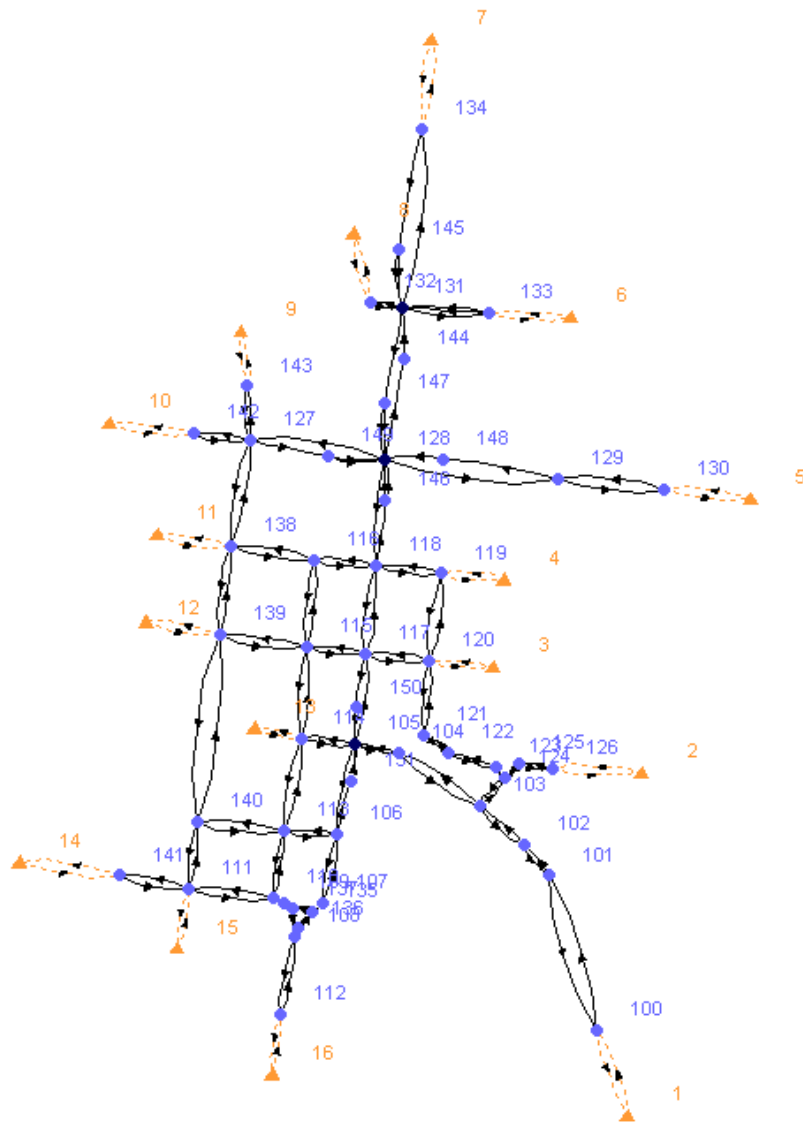
Noder representerar oftast korsningar, vilka kan kodas efter givna omständigheter; högerregeln, väjningsplikt, cirkulationsplatser eller trafiksignalreglerade korsningar. I Figur 4.1 representerar de blå cirklarna noderna i nätverket. Antal körfält in och ut från en korsning kan bestämmas, såväl som en trafiksignals gröntid respektive rödtid. Noderna anges alltid som heltal, måste vara unika, och är beroende av antalet zoner, se kapitel 4.3. Möjligheten att skapa noder som är detaljerade och kodade enligt verkliga förhållanden är stor. Ibland krävs dock två noder för att kunna representera verkligheten, eftersom en nod endast kan klara av att hantera ett visst antal korsande länkar.¹⁶

¹³ CONTRAMs hemsida, 2007-12-03

¹⁴ Kungliga tekniska högskolans hemsida, 2007-06-25

¹⁵ Kungliga tekniska högskolans hemsida, 2007-06-25

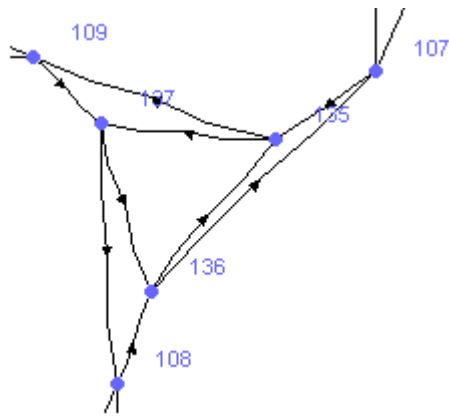
¹⁶ CONTRAM User Manual, s. 45



Figur 4.1 Nätverk ur CONTRAM över Linnégatan i Växjö med noder, länkar och zoner som beskrivs i kapitlen 4.1, 4.2 och 4.3 .

4.2 Länkar

Länkar används för att representera körfält eller svängande rörelser mellan noder som i en cirkulationsplats, se Figur 4.2. Alla länkar är enkelriktade, och de går att programmera för både vänster och högertrafik. I Figur 4.2 nedan representerar länkarna de svarta linjerna mellan noderna.



Figur 4.2 Cirkulationsplats i CONTRAM-nät.

En länk skapas genom att användaren markerar den nod som länken skall gå från och därefter den nod som länken skall sluta i. Länkens längd anges i meter och det är viktigt att zonen anknypningar har realistiska längder för att kunna representera det verkliga nätverket.¹⁷

4.3 Zoner

En zon fungerar både som ursprungs- och destinationspunkt. En zon kan potentiellt ha ett obegränsat antal anslutningar som länkar den till ett nätverk och tillåter trafik att ta sig in i, eller lämna, nätverket från flertalet olika platser. Däremot kan trafik inte passera genom zoner och vidare till andra platser i nätverket. Följaktligen kommer varje fordon som tar sig in i en zon att ha den som sin slutstation. Zoner numreras automatiskt från 1 till n, medan noder har nummer större än n. För att underlätta kan zonerna beskrivas med namn. I Figur 4.1 visas zoner som de orange trianglarna.

När en zon skapas i nätverket kommer den att ha en initial position. Positionen utgörs av dess x- och y koordinater, vilka relaterar till nätverksbakgrunden som definierades då nätverket byggdes upp från början. En kartbild kan användas som bakgrund för att underlätta uppbyggnaden av nätverket. Detta gör att zonernas placering blir mer korrekt och inte behöver ändras senare.¹⁸

4.4 Tidsintervall

I CONTRAM kan upp till 300 tidsintervall skapas, och på så sätt kan trafikintensitet och andra variabler som varierar under dygnet göras mer rättvisa i analysen. Vid tider då trafiksituationen förändras mycket kan flera kortare tidsintervall väljas, medan

¹⁷ CONTRAM User Manual, s. 40

¹⁸ CONTRAM User Manual, s. 39

längre intervall är lämpliga vid konstanta/jämna flöden. Många av programmets beräkningar bygger på tidsintervallen, varför det är viktigt att ange dessa i ett tidigt stadium. Annars måste de definieras om senare i processen och därigenom måste stora delar av analysarbetet göras om.¹⁹

¹⁹ CONTRAM User Manual, s. 35

5 Con2arcView

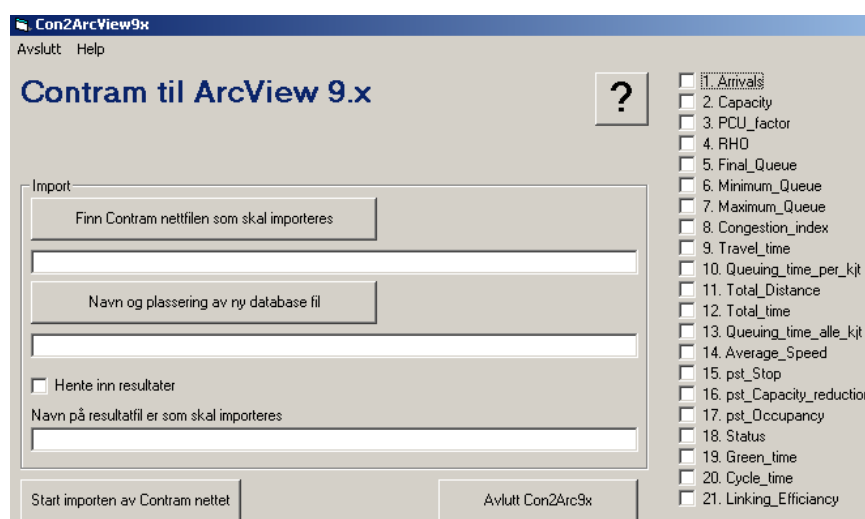
Nätverksfiler som byggts upp i trafikanalysverktyget CONTRAM kan användas i ArcMap om de först konverteras till rätt slags filformat. Genom transformationsprogrammet Con2arcView konverteras nätverksfilerna med tillhörande attribut, bundna till nätverkets länkar respektive noder, till databasformat, och kan sedan öppnas i ArcMap. I Con2arcView finns möjlighet att välja de attribut som anses vara av intresse för ändamålets analys, se Tabell 5.1. Con2arcView är framtaget av Ramböll Norge AB i Trondheim.

Tabell 5.1 Variabler som kan överföras till ArcMap med hjälp av Con2arcView.

Arrivals	Ankomster till stopplinjen eller sist i kön (antal fordon/h), fordonsflödet under tidsintervallet för specifik länk
Capacity	Effektiv kapacitet vid stopplinjen (pcu/h)
PCU-factor	Medelförhållandet av pcu till antal fordon
RHO	Ankomster/kapacitet
Final queue	Slutlig kö, i medel (antal fordon)
Minimum queue	Minsta kö, under den gröna tiden (antal fordon)
Maximum queue	Största kö, under den röda tiden (antal fordon).
Congestion index	Förhållandet av den verkliga restiden jämfört med den vid fritt flöde
Travel time	Medelrestid för fordon som ankommer under ett restidsintervall inkluderat kötid (min/fordon)
Queuing time per kjt	Medelkötiden för fordon som ankommer under ett tidsintervall (min/fordon)
Total distance	Totalt avstånd tillryggalagt av alla fordon (fordonskm)
Total time	Total tid för alla fordon (fordonsh)
Queuing time alle kjt	Total kötid för alla fordon (fordonsh)
Average speed	Medelhastighet beräknad utifrån totalt reseavstånd och tid (km/h)
Pststop, %	Procentandel av det totala antalet fordon som tvingas stanna vid stopplinjen
Pst capacity reduction, %	Procentandel av aktuell kapacitet (pga. blockering) av den normala kapaciteten
Pst occupation %	Den slutliga köns medelvärde som procent av länkens totala lagringskapacitet (max 1.0)
Status	Kod som indikerar stopp, reducerad kapacitet, samt om länken är full vid slutet av tidsintervallet
Green time	Total grön tid, från länken sett, i sekunder (sek)
Cycle time	Total cykeltid, för signalreglerade länkar (sek)
Linking efficiency	Länkförändrets effekt uttryckt som (förhållande - 1) (+ fordon eller - fordon)

5.1 Överföring av data från CONTRAM till ArcView idag

Nätverks- respektive resultatfiler från CONTRAM hämtas in i transformationsprogrammet, Con2arcView, där de attribut som är av intresse väljs, se Figur 5.1, för att sedan konverteras till en databasfil. Därefter hämtas databasen in i ArcMap och attributtabeln skapas. För att data från Con2arcView skall kunna användas i ArcMap måste attributtabeln i databasen först överföras till Excel, där behandling av data görs i ett beräkningsblad,²⁰ se kapitel 5.1.1.



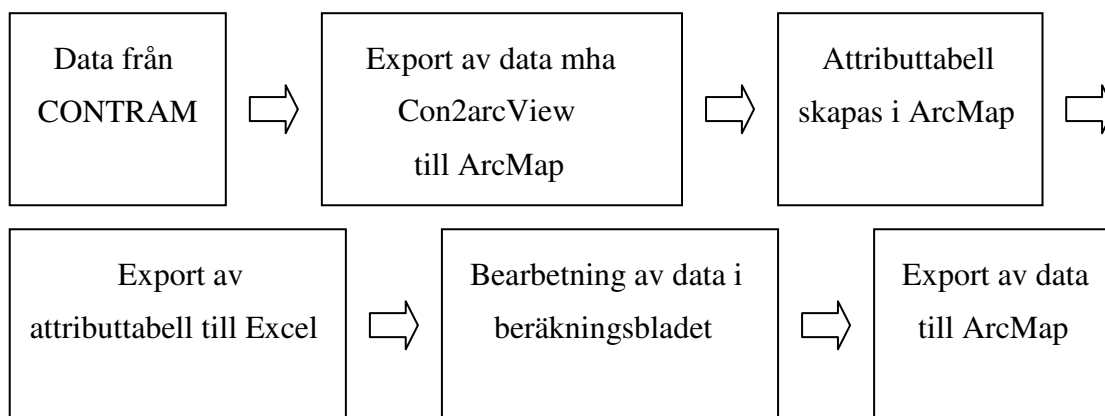
Figur 5.1 Vy över transformationsprogrammet Con2arcView.

5.1.1 Beräkningsblad

Ett beräkningsblad är framtaget av Ramböll Norge AB i Trondheim, för att bearbeta data som skall användas i ArcView. När attributtabeln är skapad i ArcMap kan användaren exportera attributtabeln till beräkningsbladet i Excel för bearbetning. Användaren markerar själv de länkar som skall analyseras, och med hjälp av beräkningsbladet beräknas det totala flödet hos varje länk som sedan summeras, se även Figur 5.2 nedan. Därefter kan den modifierade attributtabeln återigen exporteras till ArcMap för vidare bearbetning.

Export av data kan vara speciellt tidskrävande om det är stora nätverk som skall analyseras eftersom det då finns många länkar, och som tidigare nämnts har varje länk ett flöde i varje riktning, vilket betyder att excelfilerna ibland kan vara stora och svåra att hitta rätt information i.

²⁰ Anna Windal, 070817



Figur 5.2 Schematisk bild över hur överföringen av data från CONTRAM till ArcMap går till i dagsläget.

6 Visual Basic

Visual Basic är ett programmeringsspråk, en uppsättning av kommandon och kommandotillägg som används för att ge instruktioner till datorn, som idag används över hela världen.²¹ Visual Basic är skapat av företaget Microsoft, som baserade Visual Basic på programmeringsspråket BASIC, Beginner's All-purpose Symbolic Instruction Code, som skapades för nybörjare inom programmering. BASIC har funnits i mer än 35 år. Visual Basic är ett grafiskt baserat programmeringsspråk, se kapitel 6.1, medan BASIC är textbaserat.

Visual Basic är ett tolkat och kompilerat språk. Ett tolkat språk betyder att Visual Basic kan köra program efterhand som de skrivs. Ett kompilerat språk kräver ett extra steg innan programmet kan köras, det kompilerade programmet sparas i datorns egna interna språk och inte i det ursprungliga programmeringsspråket. När ett program är kompilerat är programmet snabbare och det är svårare för utomstående att ändra i koden. Genom att Visual Basic har båda dessa funktioner har programmet blivit ett av världens mest populära.²²

6.1 Visual Basic for Applications

Visual Basic for applications, VBA, är ett objektorienterat programmeringsspråk som finns inkluderat i ArcGIS Desktop och är en förenklad version av Visual Basic. Ett objektorienterat programmeringsspråk arbetar med objekt som har olika egenskaper och uppför sig på olika sätt. Många av objekten är knappar, verktyg, fönster och dialogrutor, vilka känns igen från Word, Excel och andra Windowsbaserade program.²³

6.2 Visual Basic Editor

Visual Basic Editor, hädanefter benämnt VBE, är ett verktyg för att kompilera och köra program.²⁴ I VBE skrivs all programkod som sedan kan organiseras i projekt, moduler och procedurer. De av VBEs komponenter som använts under arbetets gång beskrivs i kapitlen 6.2.1 – 6.2.5.

²¹ Perry, Greg (2002), s. 8

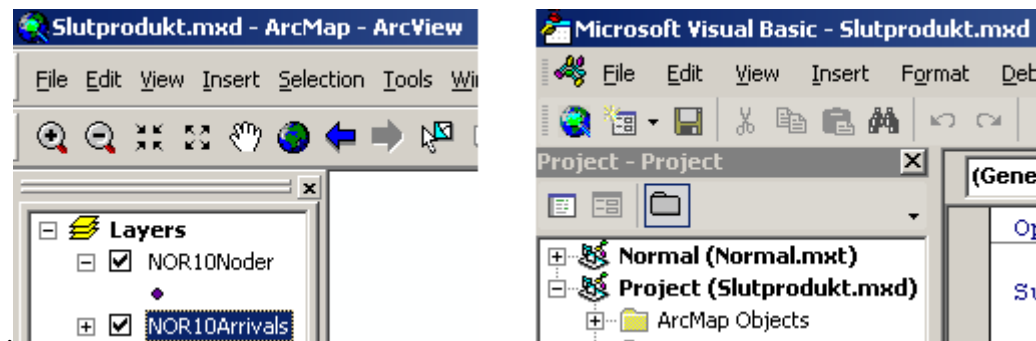
²² Perry, Greg (2002), s. 10

²³ Burke, Robert (2003), s. 1

²⁴ Burke, Robert (2003), s. 28

6.2.1 Projekt

Ett projekt är en fil där programkoden och användarkontroller (UIControl), som beskrivs vidare i kapitel 6.2.5 sparas. De projekt som programkod kan sparas i är den ursprungliga "Normal.mxt" eller i ett som är skapat av användaren, i detta fall i "Slutprodukt.mxd", se Figur 6.1.

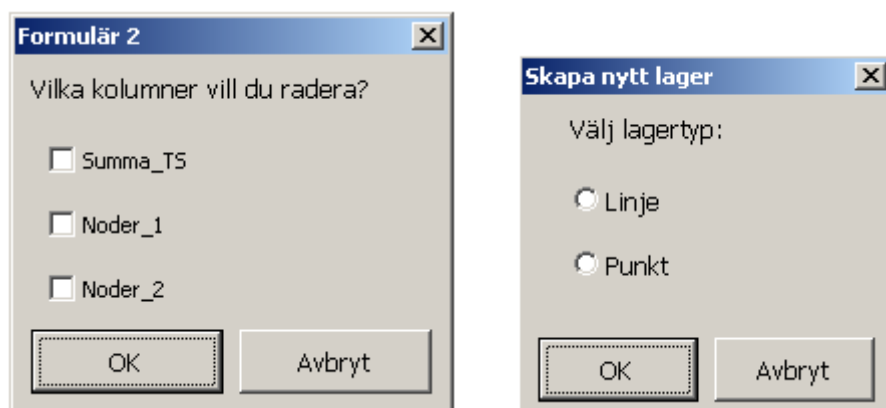


Figur 6.1 Projektet i ArcMap respektive i VBE.

När Visual Basic Editor startas öppnas projektfönstret automatiskt, och där finns alla de komponenter som dittills skapats i projektet. Projektet "Slutprodukt" är det som öppnas och används i ArcMap, och kan ses i den högra bilden i Figur 6.1.

6.2.2 Formulär

Alla dialogrutor som programmeras kallas formulär. Ett formulärs utseende kan variera beroende på vad den som skapar programmet vill att det skall användas till; vissa formulär består av endast två knappar, "OK" och "Avbryt", medan andra kan innehålla kryssrutor, punktlistor eller rullister för att användaren skall kunna mata in information i programmet, se Figur 6.2. I Bilaga 1 finns koden till formuläret "Skapa nytt lager".



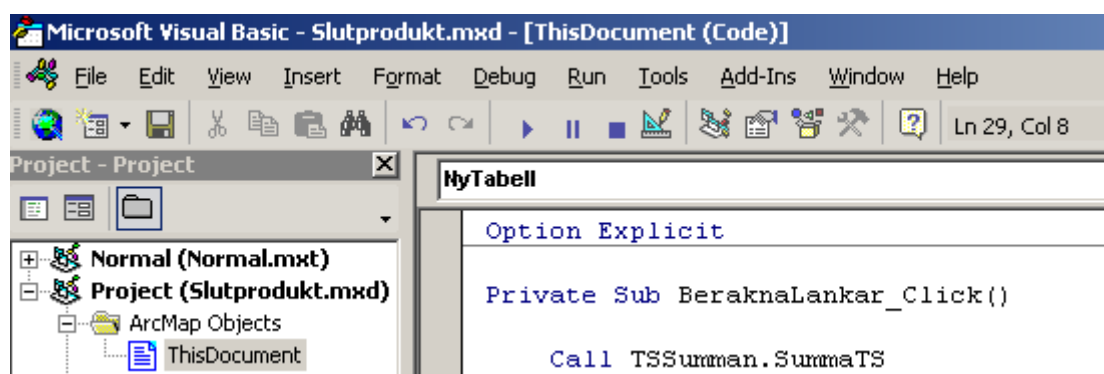
Figur 6.2 Formulär med kryssrutor respektive punklista.

6.2.3 Procedur

En procedur är ett kodblock som beskriver vad som skall utföras då proceduren körs. Oftast är det en procedur som är programkoden bakom en knapp i ArcMap. Exempel på en procedur är "Private Sub cmdOK_Click()" som finns i Bilaga 1.

6.2.4 Modul

En modul i ett projekt kan innehålla både privata och allmänna procedurer. En privat procedur kan endast anropas inifrån samma modul, medan en allmän procedur kan anropas från vilken modul som helst.²⁵ Varje projekt har en modul som heter ThisDocument, se Figur 6.3. I denna modul anropas oftast de olika kontrollerna, se kapitel 6.2.5, som behövs i ett specifikt projekt. Själva proceduren för en kontroll placeras ofta i en egen modul, se Bilaga 2.



Figur 6.3 Modulen ThisDocument

²⁵ Chang Kang-Tsung (2008), s. 16

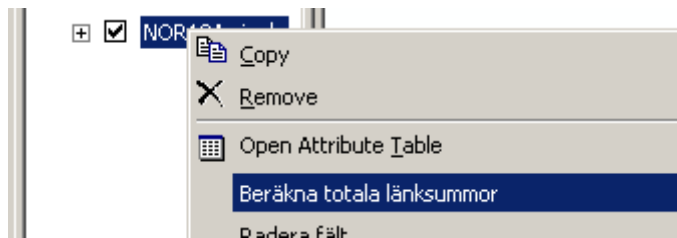
6.2.5 Användarkontroller

Användarkontroller (UIControls, User Interface Controls) är kommandon som kan användas för att skapa nya kontroller i Visual Basic. Det finns fyra olika typer:

- UIButton (knapp)
- UITool (verktyg)
- UIComboBox
- UIEditBox

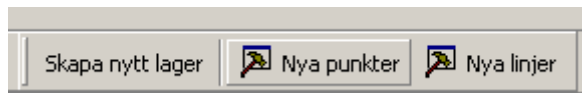
I arbetet används endast UIButton och UITool, varför de övriga två, UIComboBox och UIEditBox, inte kommer att diskuteras vidare.

UIButton – Används för att få programmet att påbörja, avsluta eller avbryta en händelse eller en rad händelser. I det här fallet är ”Beräkna totala länksummor” en UIButton, se Figur 6.4.



Figur 6.4 UIButton som får programmet att påbörja beräkningen.

UITool – Används när en interaktion med skärmen skall utföras. Verktygen ”Nya punkter” och ”Nya linjer” är exempel, se Figur 6.5.



Figur 6.5 UITools.

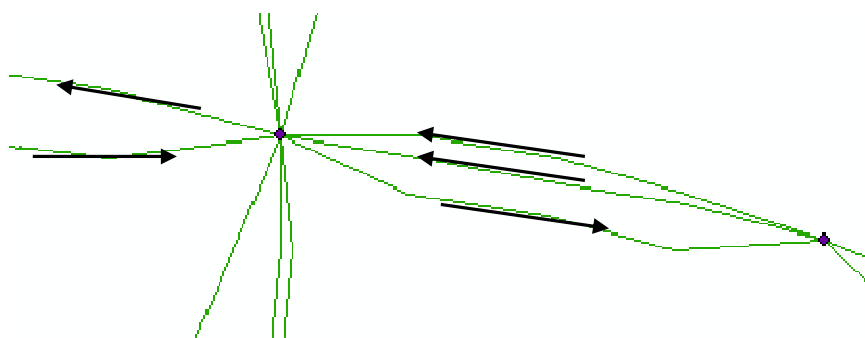
För att en användarkontroll skall fungera måste den kopplas till en kod i projektet. I koden skrivs vad som skall hända då användaren klickar på en specifik knapp.

6.3 ArcObjects

ArcObjects är en uppsättning specialdesignade datorobjekt, utvecklade av ESRI, för programmering av Desktop-GIS applikationer. De typer av objekt som ArcObjects innehåller är lager, features, tabeller, kartografiska symboler, samt vad de byggs upp av; punkter, linjer, polygoner, kolumner, färger med mera.²⁶

6.3.1 Feature class

Feature class är en samling objekt som har samma typ av geometri. I detta fall är de två shapefilerna Nor10Noder (punkter) och Nor10Arrivals (linjer) två olika feature-klasser, vilkas egenskaper sparas som tabeller.²⁷ Vid beräkningar i arbetet kommer endast feature klassen NOR10Arrivals att användas, varför ingen vikt läggs vid Nor10Noder. En linje i NOR10Arrivals representerar ena körriktningen av en dubbelriktad gata, eller ett körfält om gatan har flera sådana. Varje linje representerar således ett körfält, se Figur 6.6.



Figur 6.6 Körriktningar hos NOR10Arrivals.

6.3.2 Feature

Alla de komponenter som bygger upp en feature-klass är features. Som synes i de olika kolumnerna i tillhörande attributtabel, se Figur 7.1, är [TS1] ett exempel på en feature. Hur objektet Feature används kan ses i Bilaga 3.

²⁶ Burke, Robert (2003), s. 4

²⁷ Chang Kang-Tsung (2008), s. 2

6.3.3 Cursor

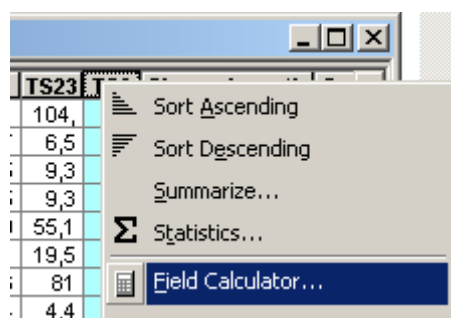
En cursor är en slags markör som har tillgång till data, och som låter ett makro, det vill säga ett eller flera kodblock, stega sig igenom ett antal rader i en tabell.²⁸ För att se hur objektet Cursor används, se Bilaga 4.

6.3.4 QueryFilter

Ett QueryFilter (frågeställningsfilter) används här tillsammans med ett WhereClause ("där"-klausul) för att rätt värden skall kunna hittas ur tabellen. Kombinationen skulle kunna se ut som följer: QueryFilter.WhereClause = "TS1 = 2". Detta betyder att när programmet sedan körs kommer det att leta efter värdet 2 i kolumn [TS1]. QueryFilter diskuteras också i kapitel 7.4. Hur objektet QueryFilter används i programkoden ses i Bilaga 5.

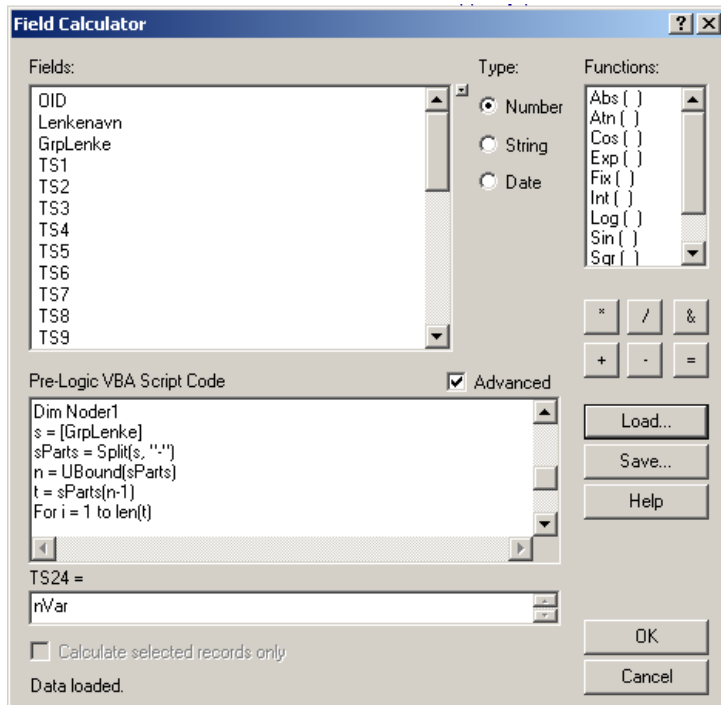
6.3.5 Calculator

Genom objektet Calculator kan beräkningar ske genom programkoden, och användaren behöver till exempel inte använda FieldCalculator (kolumnkalkylator), som finns i attributtabelen, FieldCalculator kan användas för att utföra olika beräkningar i en kolumn, och som användaren kommer åt genom högerklicksmenyn, se Figur 6.7 och 6.8. Den kod som skrivs in i FieldCalculator kan också användas i ett makro, och därifrån kopplas till en knapp så att användaren inte måste öppna attributtabelen för att utföra de önskade beräkningarna.



Figur 6.7 Tillgång till FieldCalculator fås genom högerklick i attributtabelen.

²⁸ Chang Kang-Tsung (2008), s. 70



Figur 6.8 FieldCalculator.

För att den kod som utför beräkningarna i FieldCalculator skall kunna användas i ett makro måste den skrivas som ett preExpression. I ett preExpression skrivs koden som en textsträng så att den kan läsas av kalkylatorn som sedan utför beräkningarna för en given kolumn. Slutligen är det Calculator som gör beräkningen genom att köra textsträngen som står i preExpression. Hur Calculator fungerar i programkoden ses i Bilaga 4.

6.3.6 Fields

Objektet Fields är en samling kolumner såsom en feature klass, som tidigare beskrevs i kapitel 6.3.2, eller en tabell. I båda fallen är Fields relaterat till ett Table-objekt. Ett Fields-objekt består av ett eller flera Field-objekt.²⁹

²⁹ Chang Kang-Tsung (2008), s. 69-70

7 Utförd programmering i VBA och ArcMap

För att programmet i slutändan skall beräkna det totala flödet för alla unika nodpar, krävs ett par delmoment:

- Flödet för varje tidsintervall summeras för varje länk, och sparas i en ny kolumn.
- Kolumnen [GrpLenke] delas upp i två för att separera till- och från-noder, vilka sparas i två nya kolumner.

7.1 Attributtabellen

När data exporteras som databasfil från CONTRAM till ArcMap finns all information samlad i attributtabeller. Varje shapefil i databasfilen har en egen attributtabel. I följande kapitel beskrivs attributtabellen hos linjeshapefilen NOR10Arrivals, se även Figur 7.1.

7.1.1 Kolumner hos attributtabellen för NOR10Arrivals

[Object ID]:	Ett unikt nummer för varje enskild länk.
[Shape]:	Objektets geometrityp, i det här fallet polyline (linje).
[Lenkenavn]:	Anger mellan vilka noder länken stäcker sig, samt om det finns fler länkar med samma sträckning.
[GrpLenke]:	Anger mellan vilka noder länken sträcker sig.
[Shape_Length]:	Länkens längd, det vill säga gatans längd.
[TS1 till och med TS24]:	Antalet passerande fordon under ett visst tidsintervall.

Object ID *	Shape *	Lenkenavn	GrpLenke	Shape Length	TS1	TS2	TS3	TS4	TS5	TS6
1	Polyline	1/100/1	1 - 100	99,243045	98,2	18,2	6,4	7	10,2	41,3
2	Polyline	2/126/1	2 - 126	96,888	6,1	1,1	0,4	0,4	0,6	2,6
3	Polyline	3/120/1	3 - 120	69,889588	8,7	1,6	0,6	0,6	0,9	3,7
4	Polyline	4/119/1	4 - 119	68,728121	8,7	1,6	0,6	0,6	0,9	3,7
5	Polyline	5/130/1	5 - 130	95,402836	51,7	9,6	3,4	3,7	5,4	21,8
6	Polyline	6/133/1	6 - 133	92,775489	18,3	3,4	1,2	1,3	1,9	7,7
7	Polyline	7/134/1	7 - 134	96,659427	76,1	14,1	4,9	5,5	7,9	32,1
8	Polyline	8/132/1	8 - 132	77,333414	4,2	0,8	0,3	0,3	0,4	1,8

Figur 7.1 Kolumner hos attributtabelen för NOR10Arrivals

7.1.2 Skapade kolumner i attributtabelen för NOR10Arrivals

[Summa_TS]: Summan av TS1 – TS24

[Noder_1]: Frånnod

[Noder_2]: Tillnod

[Lanksumma]: Summan av det antal fordon som rör sig mellan två specifika noder.

7.2 Beräkning av det totala flödet hos varje enskild länk

Proceduren beräknar det totala fordonsflödet hos en länk. Makrot "SummaTS" summerar det totala fordonsflödet hos en länk under de givna 24 tidsintervallen. Först skapas en kolumn, [Summa_TS], där de beräknade summorna för varje länk skall sparas. Med hjälp av en cursor adderas flödena successivt för en länks respektive tidsintervall, och sparas sedan på motsvarande plats i [Summa_TS],³⁰ se Bilaga 3.

7.3 Skapandet av Noder 1 och Noder 2

Koden delar kolumnen [GrpLenke], se Figur 7.1, i två nya kolumner, [Noder1] och [Noder2], se Figur 7.2.

³⁰ Burke, Robert (2003)

	TS21	TS22	TS23	TS24	Summa TS	Shape Length	Noder 1	Noder 2	Lanksumma
▶	271,5	220,1	104,6	0	7118	99,243045	1	100	14147
▶	16,9	13,7	6,5	0	442	96,888	2	126	884
▶	24,1	19,5	9,3	0	632	69,889588	3	120	1264
▶	24,1	19,5	9,3	0	632	68,728121	4	119	1264
▶	143	115,9	55,1	0	3748	95,402836	5	130	7502
▶	50,6	41	19,5	0	1326	92,775489	6	133	2652
▶	210,3	170,6	81	0	5515	96,659427	7	134	11115
▶	11,5	9,4	4,4	0	303	77,333414	8	132	606

Record: 1 Show: All Selected Records (0 out of 156 Selected)

Figur 7.2 Kolumnerna [Noder1] och [Noder2].

I preExpression-delen av de två makrona "Noder1" och "Noder2" har funktionen Split använts för att kunna dela upp [GrpLenke] i två nya kolumner, [Noder_1] och [Noder_2]. Värdena i [GrpLenke] konverteras först till textsträngar för att Splitfunktionen skall kunna "klippa av" rätt delar av kolumnens innehåll. Genom att i Splitfunktionen ange att textsträngarna skall delas vid tecknet "-", skapas nu de nya kolumnerna med sina höger- respektive vänsterdelar av [GrpLenke],³¹ se Figur 7.2 samt Bilaga 4.

7.4 Beräkning av det totala flödet mellan ett nodpar

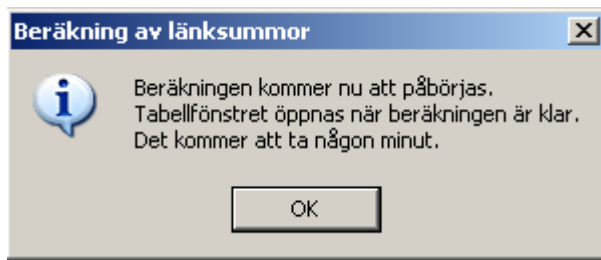
För att beräkna det totala flödet mellan ett nodpar måste programmet leta igenom [Noder_1] och [Noder_2] efter matchande par. De flesta länkar förekommer som par och har kombinationer som 1 - 100 respektive 100 - 1, men det händer att en nodparskombination förekommer fler än en gång, vilket gör det svårare att skriva koden för makrot.

Att välja ut värden ur de kolumner som användaren finner intressanta kan lätt göras genom en SQL-sökning (Standard Query Language), som hittas i rullistmenyn under Selection i ArcMaps huvudmeny. Samma slags SQL kan också användas i ett makro, men måste då skrivas tillsammans med ett QueryFilter.

När användaren klickar på "Beräkna totala länksummor" i högerklicksmenyn, kommer först procedurerna "SummaTS" samt "Noder1" och "Noder2" att anropas innan slutligen "BeraknaTotLanksumma" anropas och körs, se Bilaga 5.

Innan beräkningen i proceduren "BeraknaTotLanksumma" påbörjas visas ett informationsmeddelande, se Figur 7.3. Detta för att användaren skall veta vad som händer. När användaren klickar på "OK"-knappen startar beräkningen.

³¹ESRI Support Centers Hemsida, 2007-10-10



Figur 7.3 Informationsmeddelande.

För att kunna loopa igenom tabellen efter nodpar som matchar varandra har tre features och tre cursors skapats i makrot "BeraknaTotLanksumma", se Bilaga 5:

- Cursor1 letar igenom Feature1 efter nya värden för xVal1 och xVal2.
- Cursor2 letar igenom Feature2 efter värden i [Noder_1] och [Noder_2] som skall stämma överens med villkoret i WhereClause för QueryFilter.
- Cursor3 letar igenom Feature3 efter värden i [Noder_1] och [Noder_2] som skall stämma överens med villkoret i WhereClause för QueryFilter2.

När de matchande nodparen hittas genom WhereClause1 summeras deras [Summa_TS]-värden efter hand i variabeln Sum. WhereClause2 har samma villkor som WhereClause1, vilket betyder att samma matchande nodpar som tidigare hittas, åter påträffas. Denna gång sparas det slutliga Sum-värdet på de matchande nodparens rader i [Lanksumma]³².

7.5 Öppna attributtabeln

"Tabellen" är ett makro som öppnar upp attributtabeln. För att användaren skall veta när beräkningen av den totala länksumman är klar öppnas attributtabeln där de beräknade värdena lagts till. Makrot "Tabellen", se Bilaga 6, anropas i slutet av "BeraknaTotLanksumma",³³ se Bilaga 5.

7.6 Radera fält

När användaren klickar på "Beräkna totala länksummor" kontrolleras om de tabeller som kommer att skapas under programmets gång redan finns. Om de finns kommer programmet att säga ifrån och avbrytas (för idéer om förbättringar se kapitel 9.3). För att användaren inte skall behöva öppna upp attributtabeln och radera varje kolumn

³² ESRI Support Centers Hemsida, 2007-10-19

³³ ESRI Support Centers Hemsida, 2007-11-07

för sig skapas ett makro, "RaderaField", som raderar kolumnerna [Noder_1], [Noder_2], [Summa_TS] och [Lanksumma] samtidigt,³⁴ se Bilaga 7.

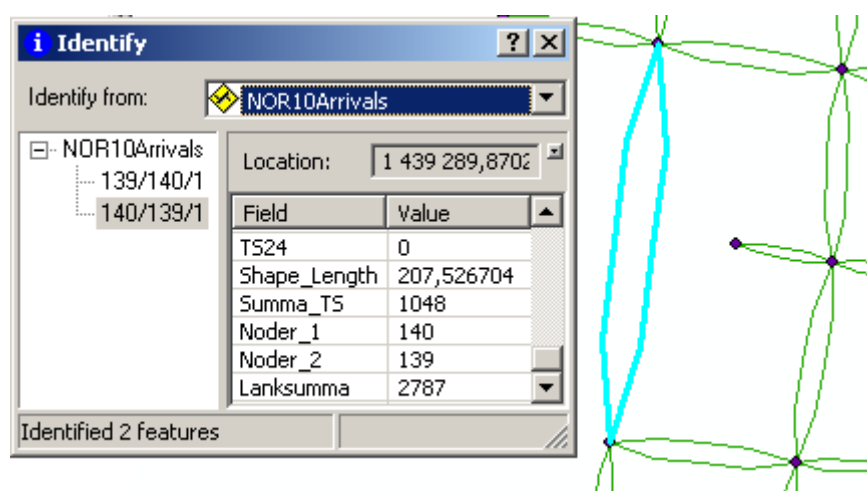
³⁴Chang Kang-Tsung (2008), s. 78

8 Resultat och slutsatser

8.1 Resultat

Genom den programmering som utförts under studien kan ArcMap utföra nya beräkningar. Dessa används genom ett klick på ”Beräkna totala länksummor” i högerklicksmenyn hos kartlagret för ankomster, se Figur 8.1. Genom ett klick på ”Beräkna totala länksummor” beräknas det totala flödet hos respektive länk i ett specifikt nätverk. Användaren kan enkelt få uppgifter om hur stort flödet är hos länkarna i nätverket, antingen genom attributtabellen som öppnas när beräkningen är klar, eller via informationsläget som beskrivs nedan.

I det slutgiltiga nätverket får användaren nu även uppgifter om det totala flödet hos ett nodpar. Genom att i informationsläget klicka på en länk mellan ett nodpar fås det totala flödet hos båda länkarna, se Figur 8.1.



Figur 8.1 Informationsläge i ArcMap.

8.2 Slutsatser

De nya kommandon som skapats kommer att hjälpa användaren när det gäller att behandla data som importerats från Con2arcView till ArcMap. Antalet moment som tidigare krävdes vid överföringen av data till ArcMap har minskats.

När användaren arbetar med, speciellt stora, nätverk som innehåller många länkar, finns risk att fel länkpar paras ihop i beräkningsbladet, som beskrivs i kapitel 5.1.1. Risken för fel i slutlig data har reducerats eftersom användaren inte längre själv behöver para ihop matchande länkpar i beräkningsbladet och sedan exportera resultatet till ArcMap.

Tidsåtgången som krävs vid exportering av data med hjälp av Con2arcView beror på hur stora de nätverk som skall behandlas är. Därför kommer det vara mer effektivt att

använda de nya kommandona på större nätverk, men de är även användbara vid mindre nätverk. Även om det tar en stund för programmet att utföra beräkningarna, har antalet arbetsmoment minskats när det gäller att kunna använda CONTRAM-data i ArcMap

Användarvänligheten vid importering av data från CONTRAM till ArcMap har förbättrats genom att användaren nu endast exporterar data från CONTRAM till ArcMap med hjälp av Con2arcView och därefter klickar på ”Beräkna totala länksummor”.

”Beräkna totala länksummor” utökar även möjligheterna för ingenjörer att enklare överskådliggöra de resultat som utvunnits ur ett trafikanalysprogram. På så sätt ges beslutsfattare en bättre möjlighet att se helheten i en trafikanalyssituation, och därmed lättare och snabbare kunna fatta rätt beslut.

9 Förslag till fortsatt arbete

Genom att skapa fler nya programdelar samt arbeta vidare med dem som skapats kan trafikplaneringsarbetet förenklas ytterligare.

9.1 Skapa nytt lager

När användaren klickar på knappen ”Skapa nytt lager”, visas ett formulär där användaren får välja om det nya lagret skall vara ett punkt- eller ett linjelager, se Figur 6.2. I det fall användaren väljer ”Punktlager” och sedan klickar ”OK” kommer proceduren ”NyttPunktlager” att köras och ett nytt punktlager skapas, se Bilaga 1.

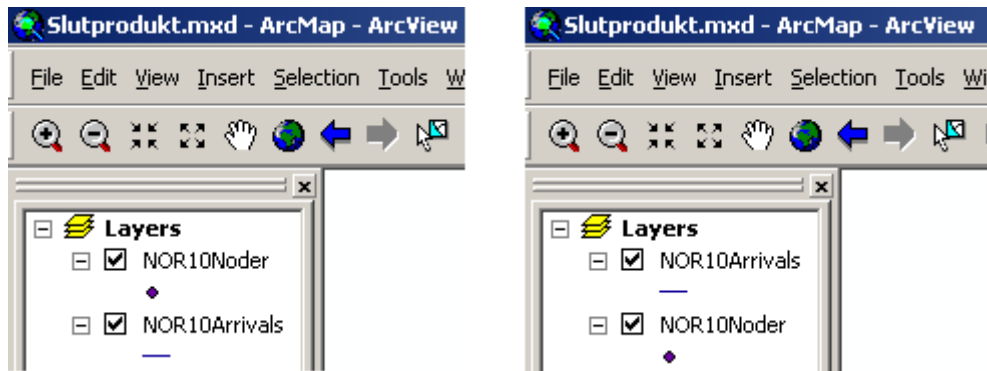
Om användaren istället önskar ett nytt linjelager, klickar han eller hon först i formulärets val för ”Linjelager”, och därefter ”OK”, varpå proceduren ”NyttLinjelager” anropas och körs, se Bilaga 2, vilket resulterar i ett nytt linjelager. Det är bra att skapa ett nytt lager innan verktygen ”Nya linjer” respektive ”Nya punkter” aktiveras, eftersom dessa då inte kommer i konflikt med ursprungliga länk- och nodlager.³⁵

9.2 Skapa nya punkter

”Nya punkter” är ett verktyg som låter användaren skapa just nya punkter. För att verktyget skall fungera måste det översta lagret i innehållsförteckningen, se Figur 9.1, vara ett punktlager, förslagsvis ett som skapats genom ”Skapa nytt lager”. När användaren klickar på ”Nya punkter” aktiveras verktyget genom att proceduren ”SkapaNyaPunkter” körs, se Bilaga 8, och användaren kan genom vänsterklick skapa en ny punkt. Förslag till användningsområde för detta verktyg tas upp i kapitel 9.2³⁶.

³⁵ ESRI Support Centers Hemsida, 2007-09-26

³⁶ ESRI Support Centers Hemsida, 2007-10-02



Figur 9.1 Punkt- respektive linjelager som översta lager.

9.3 Skapa nya linjer

Verktyget ”Nya linjer” ger användaren möjlighet att själv skapa en eller flera nya linjer. Eftersom det är ett verktyg krävs interaktion med skärmen som användaren står för. För att verktyget skall fungera måste det översta lagret i innehållsförteckningen, se Figur 9.1, vara ett linjelager, förslagsvis ett som skapats genom ”Skapa nytt lager”. När användaren klickar på ”Nya linjer” aktiveras verktyget genom att proceduren ”SkapaNyaLinjer” körs, se Bilaga 8, och användaren kan genom vänsterklick skapa linjens brytpunkter. Linjen färdigställs genom ett dubbelklick. Förslag till användningsområde för detta verktyg tas upp i kapitel 9.6.³⁷

9.4 Skapa ny tabell

Makrot ”Ny”, se Bilaga 9, skapar en tom fristående tabell i innehållsförteckningen. När makrot anropas från ThisDocument skapas en tom tabell. Användaren kan ange antal kolumner och kolumnnamn i koden för makrot. Förslag till användningsområde diskuteras i kapitel 9.1.³⁸

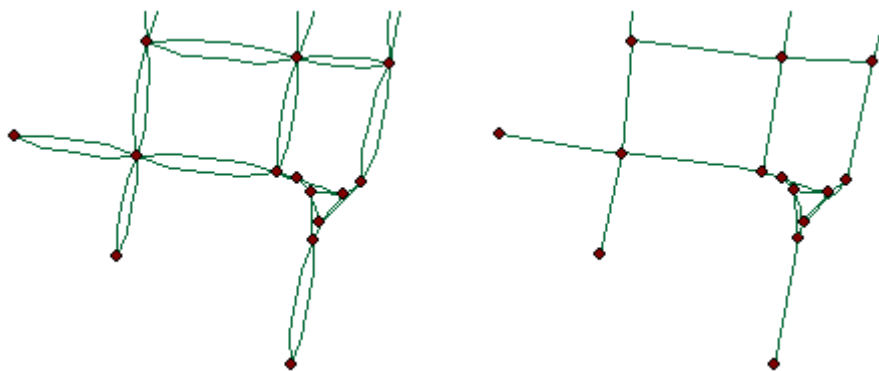
9.5 Utveckling av länknätverk och attributtabell

Efter att knappen ”Beräkna totala länksummor” klickats visar alla länkar mellan två specifika noder det sammanlagda flödet. Om antalet länkar mellan ett specifikt nodpar skulle kunna begränsas till en enda som visar det totala flödet skulle nätverket bli mer överskådligt och enkelt att hitta i. Detsamma skulle gälla för den tillhörande attributtabellen.

³⁷ ESRI Support Centers Hemsida, 2007-10-02

³⁸ ESRI Support Centers Hemsida, 2007-11-14

Genom att modifiera den befintliga koden som används i "Beräkna totala länksummor" kan flödet mellan varje nodpar skrivas till endast en utav de nu två eller tre befintliga länkarna mellan ett nodpar. Se förslag i Figur 9.2. De totala länksummorna kan sedan, tillsammans med nodpar och övriga intressanta parametrar, exporteras från attributtabeln till en tom tabell som skapas genom makrot "Ny". Se Bilaga 9.



Figur 9.2 Nätverk i ArcMap före respektive efter förslaget om förbättring.

9.6 Förslag till användningsområden för "Nya linjer" och "Nya punkter"

Verktygen "Nya linjer" och "Nya punkter", som beskrivs i kapitel 9.3 respektive 9.4, kan användas då användaren vill veta flöden hos specifika länkar. Dessa verktyg används i anslutning till "Skapa Nytt lager". Användaren kan markera en intressant länk med en ny punkt i ett nytt punktlager eller med en ny linje i ett nytt linjelager. Genom att göra en "Selection by location", det vill säga ett urval genom placering, kan användaren få information om de länkar som korsas av de nya punkterna eller linjerna. Urvalet kan sedan exporteras till en ny tabell som skapats genom att makrot "Ny" körts. På så vis får användaren en tabell som endast innehåller de länkar som är av intresse.

9.7 Övriga förbättringar

I rutinerna för "SummaTS", "Noder1", "Noder2" och "TotLänksumma" skapas kolumner för motsvarande värden. I det fall en kolumn redan existerar då en rutin körs, skulle antingen programkoden kunna radera den existerande kolumnen, alternativt skriva över de värden som redan finns i kolumnen.

10 Referenser

Tryckta källor

Eklundh Lars (RED) (2003), Geografisk informationsbehandling, Byggeforskningsrådet Stockholm, ISBN: 91-540-5904-6

ESRI, (1999), Getting to know ArcView GIS, Environmental Systems Research Institute, Inc, Redlands, California USA, ISBN: 1-879102-46-3

Burke Robert (2003), Getting to know ArcObjects, Environmental Systems Research Institute, Inc, Redlands, California USA, ISBN: 1-58948-018-X

Perry Greg (2000), Lär dig Visual Basic 6 på 3 veckor, Ait Falun, ISBN: 91-636-0523-6

Chang Kang-Tsung (2008), Programming ArcObjects with VBA: a task-oriented approach, Chapman & Hall, USA, ISBN: 0-8493-9283-7

Elektroniska källor

CONTRAMs manual, CONTRAM User Manual for CONTRAM 8.3

Environmental Systems Research Institute,
<http://www.esri.se/templates/Page.aspx?id=130b>. Hämtat 2007-06-15

Environmental Systems Research Institute,
<http://www.esri-sgroup.se/templates/Page.aspx?id=188>. Hämtat 2007-06-15

Environmental Systems Research Institute,
<http://www.esri-sgroup.se/templates/Page.aspx?id=556#arcmap>. Hämtat 2007-06-18

Environmental Systems Research Institute,
<http://www.esri-sgroup.se/templates/Page.aspx?id=449>. Hämtat 2007-11-26

ESRI ArcGIS Developer Online,
http://edndoc.esri.com/arcobjects/9.0/default.asp?url=/arcobjects/9.0/Samples/Tables/Open_a_table_window.htm. Hämtat 2007-11-07.

Future position X: <http://www.fpx.se/default2.asp?pageid=63>. Publicerat 2007.
Hämtat 2007-06-12

G. Venkata Vijay Kumar (2004), ESRI Support Center,
<http://forums.esri.com/Thread.asp?c=93&f=992&t=133670&mc=10#387012>.
Publicerat 2004-08-05. Hämtat 2007-11-14.

Hitchen, Miles (2006), ESRI Support Center,
<http://forums.esri.com/Thread.asp?c=93&f=993&t=197349>. Publicerat 2006-08-07.
Hämtat 2007-09-26.

Lind, Gunnar (1998), *Test av dynamiska trafiksimuleringsmodeller*, CTR,
http://www.infra.kth.se/ctr/publikationer/ctr1998_10.pdf. Publicerat 1998-07. Hämtat
2007-06-25.

MacDonald Mott and TRL, <http://www.contram.com>. Publicerat 2007-12-03.
Hämtat 2007-12-03.

R. Liviu (2002), ESRI Support Center,
<http://forums.esri.com/Thread.asp?c=93&f=992&t=40702#151813>. Publicerat 2002-
03-21. Hämtat 2007-10-02.

Tapani, Stefan (2003), *Trafikledning för bättre miljö*. Hämtat från Statens Väg- och
Transportforsknings Institut:s hemsida,
<http://www.vti.se/EPiBrowser/Publikationer%20-%20English/M954.pdf>.
Publicerat 2003 . Hämtat 2007-06-20

Van Raalte Suanne, Chalmers tekniska högskola,
http://www.design.chalmers.se/projects/histgis/virtuelltkulturarv/vk_txt.html#mot%20mer. Publicerat 2000-11-30. Hämtat 2007-11-25.

Vasanthi S. (2002), ESRI Support Center,
<http://forums.esri.com/Thread.asp?c=93&f=992&t=70982&mc=3&ESRISessionID=gZ7bNzwkyZ7ByqJJp6AjND2VUirysZYAXuZqAGq04n3a4IBFLwGZTOo96ZIDZH%2DrwF0nlw%3D%3D#187162>. Publicerat 2002-08-21. Hämtat 2007-10-19.

Walkup, Henry (2003), ESRI Support Center,
<http://forums.esri.com/Thread.asp?c=93&f=992&t=92292&mc=4#msgid257192>.
Publicerat 2003-05-08. Hämtat 2007-10-10.

Öhman, Jeanette (2000), *GIS inom transportbranchen*, Center for Traffic Research,
Kungliga tekniska högskolan,
http://www.infra.kth.se/GIS/publications/msc_theses/pdf/JOe-Final.pdf.
Publicerat: 2000-09. Hämtat 2007-11-26.

Bilder

Figur 2.2 GIS-centrum, Lunds universitet,
<http://www.giscentrum.lu.se/vadargis.htm>.
Publicerat 2003-08-29. Hämtat 2007-12-03

Övriga bilder är gjorda av författarna.

11 Bilagor

Bilaga 1	frmNyttlager
Bilaga 2	NyaLager
Bilaga 3	TSSumman
Bilaga 4	DelaUppLank
Bilaga 5	TotLanksumma
Bilaga 6	OppnaTabellen
Bilaga 7	RaderaFields
Bilaga 8	ThisDocument

Bilaga 1 frmNyttLager

Option Explicit

```
Private Sub cmdOK_Click()
```

```
'****
```

```
Visar formuläret för nytt lager, som innehåller en punktlista med valmöjligheterna Punkt och Linje. När OK klickas anropas, beroende på om Punkt eller Linje valts, antingen NyaLager.NyttPunktLager eller NyaLager.NyttLinjeLager.
```

```
'****
```

```
Dim StrMedd As String
```

```
If ((optPunkt.Value = False) And (optLinje = False)) Then  
    StrMedd = MsgBox("Du måste välja ett alternativ.", _  
        vbExclamation, "Fel!")
```

```
ElseIf (optPunkt.Value = True) Then  
    frmNyttLager.Hide  
    Call NyaLager.NyttPunktLager
```

```
Else  
    frmNyttLager.Hide  
    Call NyaLager.NyttLinjeLager  
End If
```

```
End Sub
```

```
Private Sub cmdAvbryt_Click()
```

```
'****
```

```
Stänger formuläret NyttLager då kommandoknappen Avbyt klickas.
```

```
'****
```

```
    frmNyttLager.Hide  
    optPunkt.Value = False  
    optLinje.Value = False
```

```
End Sub
```

Bilaga 2 NyaLager

Option Explicit

```
Public Sub NyttLinjelager()
```

```
'****
```

```
Skapar ett nytt tomt linjelager. Användaren måste själv fylla i var  
det nya lagret skall sparas, samt namnge det.
```

```
'****
```

```
'InputBoxar där användaren anger var den nya shapefilen ska  
'sparas, resp. dess namn
```

```
Dim Path1 As String  
Path1 = InputBox("Ange sökväg")
```

```
Dim Name1 As String  
Name1 = InputBox("Ange namn")
```

```
' Öppnar feature arbetsytan för shapefilen, dvs en mapp  
Dim pWrkspcFact As IWorkspaceFactory  
Set pWrkspcFact = New ShapefileWorkspaceFactory
```

```
Dim pFtrWrkspc As IFeatureWorkspace  
Set pFtrWrkspc = pWrkspcFact.OpenFromFile(Path1, 0)
```

```
' Skapa ID- och geometrifält för den nya featureklassen  
Dim pFldsEdit As IFieldsEdit  
Set pFldsEdit = New Fields
```

```
Dim pFldEdit As IFieldEdit  
Set pFldEdit = New Field
```

```
With pFldEdit  
    .Name = "Id"  
    .Type = esriFieldTypeInteger  
End With
```

```
pFldsEdit.AddField pFldEdit
```

```
Set pFldEdit = New Field  
Dim pGeomDefEdit As IGeometryDefEdit  
Set pGeomDefEdit = New GeometryDef
```

```
With pGeomDefEdit  
    .GeometryType = esriGeometryPolyline  
    Set .SpatialReference = New UnknownCoordinateSystem  
End With
```

```
With pFldEdit  
    .Name = "Shape"  
    .Type = esriFieldTypeGeometry  
    Set .GeometryDef = pGeomDefEdit  
End With
```

```
pFldsEdit.AddField pFldEdit
```

```

'Skapar den nya featureklassen and skaffar en referens till den
Dim pFtrCls As IFeatureClass
Set pFtrCls = pFtrWrkspc.CreateFeatureClass(Name1, pFldsEdit,
        Nothing,Nothing, esriFTSimple, "Shape", "")

' Skapar ett nytt featurelager lägger till det till kartan
Dim pFtrLyr As IFeatureLayer
Set pFtrLyr = New FeatureLayer

Set pFtrLyr.FeatureClass = pFtrCls
pFtrLyr.Name = Name1

Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument
pMxDoc.FocusMap.AddLayer pFtrLyr
pMxDoc.ContentView(0).Refresh Nothing

```

End Sub

```
Public Sub NyttPunktlager()
```

```
'****
```

```
Skapar ett nytt tomt punktlager. Användaren måste själv fylla i var
det nya lagret skall sparas, samt namnge det.
```

```
'****
```

```
Dim Path As String
Path = InputBox("Ange sökväg")
```

```
Dim Name As String
Name = InputBox("Ange namn")
```

```
'Öppnar feature arbetsytan för shapefilen, dvs en mapp
Dim pWrkspcFact As IWorkspaceFactory
Set pWrkspcFact = New ShapefileWorkspaceFactory
```

```
Dim pFtrWrkspc As IFeatureworkspace
Set pFtrWrkspc = pWrkspcFact.OpenFromFile(Path, 0)
```

```
' Skapa ID- och geometrifält för den nya featureklassen
Dim pFldsEdit As IFieldsEdit
Set pFldsEdit = New Fields
```

```
Dim pFldEdit As IFieldEdit
Set pFldEdit = New Field
```

```
With pFldEdit
    .Name = "Id"
    .Type = esriFieldTypeInteger
End With
```

```
pFldsEdit.AddField pFldEdit
```

```
Set pFldEdit = New Field
Dim pGeomDefEdit As IGeometryDefEdit
Set pGeomDefEdit = New GeometryDef
```

```
With pGeomDefEdit
```

```
        .GeometryType = esriGeometryPoint
        Set .SpatialReference = New UnknownCoordinateSystem
    End with
```

```
With pFldEdit
    .Name = "Shape"
    .Type = esriFieldTypeGeometry
    Set .GeometryDef = pGeomDefEdit
End with
```

```
pFldsEdit.AddField pFldEdit
'Skapar den nya featureklassen och skaffar en referens till den
Dim pFtrCls As IFeatureClass
Set pFtrCls = pFtrWrkspc.CreateFeatureClass(Name, pFldsEdit,
        Nothing,Nothing, esriFTSimple, "Shape", "")
```

```
' Skapar ett nytt featurelager lägger till det till kartan
Dim pFtrLyr As IFeatureLayer
Set pFtrLyr = New FeatureLayer
```

```
Set pFtrLyr.FeatureClass = pFtrCls
pFtrLyr.Name = Name
```

```
Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument
pMxDoc.FocusMap.AddLayer pFtrLyr
pMxDoc.ContentView(0).Refresh Nothing
```

End Sub

Bilaga 3 TSSumman

Option Explicit

```
Public Sub SummaTS()
```

```
'****
```

```
Summerar för varje rad alla värden fr.o.m. TS1 t.o.m. TS24 och  
lägger till dem i en ny kolumn; SummaTS.
```

```
'****
```

```
'Deklarerar och fastställer en variabel för kartdokumentet  
Dim pMxDoc As IMxDocument  
Set pMxDoc = ThisDocument
```

```
'Deklarerar en IFeatureLayer-variabel och fastställer den med  
'ContextItem-egenskap  
Dim pFLayer As IFeatureLayer  
Set pFLayer = pMxDoc.ContextItem
```

```
'Deklarerar och fastställer en variabel för lagrets feature-  
'klass  
Dim pFClass As IFeatureClass  
Set pFClass = pFLayer.FeatureClass
```

```
'Deklarerar och fastställer variabler för Fields-kollektionen  
Dim pFields As IFields  
Set pFields = pFClass.Fields
```

```
'Deklarerar och fastställer en heltalsvariabel som ska bevara _  
SummaTS-fältets indexpositionsnummer  
Dim intPosSummaTS As Integer  
intPosSummaTS = pFields.FindField("Summa_TS")
```

```
Dim StrMedd As String
```

```
'Påbörjar If Then för att se om fältet redan finns  
If (intPosSummaTS = 1) Then  
    StrMedd = MsgBox("Fältet 'Summa_TS' finns redan!",  
                    vbExclamation)  
Exit Sub
```

```
Else
```

```
'Deklarerar och fastställer variabel för att skapa nytt fält  
Dim pFieldEdit As IFieldEdit  
Set pFieldEdit = New Field
```

```
'Ange egenskaper för fältets namn och typ  
pFieldEdit.Name = "Summa_TS"  
pFieldEdit.Type = esriFieldTypeInteger
```

```
'Använd AddFieldmetod för att lägga till fält till tabellen  
pFClass.AddField pFieldEdit
```

```
'Återställer variabeln som innehåller SummaTS-fältets index-  
'positionsnummer  
intPosSummaTS = pFields.FindField("Summa_TS")
```

End If

'Deklarerar och fastställer en heltalsvariabel för att bevara
'TS1-TS24 fältens indexpositionsnummer

Dim intPostS1 As Integer
intPostS1 = pFields.FindField("TS1")

Dim intPostS2 As Integer
intPostS2 = pFields.FindField("TS2")

Dim intPostS3 As Integer
intPostS3 = pFields.FindField("TS3")

Dim intPostS4 As Integer
intPostS4 = pFields.FindField("TS4")

Dim intPostS5 As Integer
intPostS5 = pFields.FindField("TS5")

Dim intPostS6 As Integer
intPostS6 = pFields.FindField("TS6")

Dim intPostS7 As Integer
intPostS7 = pFields.FindField("TS7")

Dim intPostS8 As Integer
intPostS8 = pFields.FindField("TS8")

Dim intPostS9 As Integer
intPostS9 = pFields.FindField("TS9")

Dim intPostS10 As Integer
intPostS10 = pFields.FindField("TS10")

Dim intPostS11 As Integer
intPostS11 = pFields.FindField("TS11")

Dim intPostS12 As Integer
intPostS12 = pFields.FindField("TS12")

Dim intPostS13 As Integer
intPostS13 = pFields.FindField("TS13")

Dim intPostS14 As Integer
intPostS14 = pFields.FindField("TS14")

Dim intPostS15 As Integer
intPostS15 = pFields.FindField("TS15")

Dim intPostS16 As Integer
intPostS16 = pFields.FindField("TS16")

Dim intPostS17 As Integer
intPostS17 = pFields.FindField("TS17")

Dim intPostS18 As Integer
intPostS18 = pFields.FindField("TS18")

```

Dim intPosTS19 As Integer
intPosTS19 = pFields.FindField("TS19")

Dim intPosTS20 As Integer
intPosTS20 = pFields.FindField("TS20")

Dim intPosTS21 As Integer
intPosTS21 = pFields.FindField("TS21")

Dim intPosTS22 As Integer
intPosTS22 = pFields.FindField("TS22")

Dim intPosTS23 As Integer
intPosTS23 = pFields.FindField("TS23")

Dim intPosTS24 As Integer
intPosTS24 = pFields.FindField("TS24")

'Deklarerar och fastställer en variabel för att skapa en feature
'cursor
Dim pFCursor As IFeatureCursor
Set pFCursor = pFClass.Update(Nothing, True)

'Deklarerar en IFeature-variabel och fastställer den med
'cursorns
'NextFeaturemetod
Dim pFeature As IFeature
Set pFeature = pFCursor.NextFeature

'Påbörja en Do Until loop som körs så tills det inte finns fler
'features i cursorn
Do Until pFeature Is Nothing
    'Använd Value property för att fastställa den första
    'features SummaTS lika med summan av värdena från TS1 tom
    'TS24
    pFeature.Value(intPosSummaTS) = _
        pFeature.Value(intPosTS1) + _
        pFeature.Value(intPosTS2) + _
        pFeature.Value(intPosTS3) + _
        pFeature.Value(intPosTS4) + _
        pFeature.Value(intPosTS5) + _
        pFeature.Value(intPosTS6) + _
        pFeature.Value(intPosTS7) + _
        pFeature.Value(intPosTS8) + _
        pFeature.Value(intPosTS9) + _
        pFeature.Value(intPosTS10) + _
        pFeature.Value(intPosTS11) + _
        pFeature.Value(intPosTS12) + _
        pFeature.Value(intPosTS13) + _
        pFeature.Value(intPosTS14) + _
        pFeature.Value(intPosTS15) + _
        pFeature.Value(intPosTS16) + _
        pFeature.Value(intPosTS17) + _
        pFeature.Value(intPosTS18) + _
        pFeature.Value(intPosTS19) + _
        pFeature.Value(intPosTS20) + _
        pFeature.Value(intPosTS21) + _

```

```
pFeature.Value(intPosTS22) + _  
pFeature.Value(intPosTS23) + _  
pFeature.Value(intPosTS24)
```

```
'Använder UpdateFeature-metoden för att skriva det nya  
'värdet till featureklassen  
pFCursor.UpdateFeature pFeature  
'Använd NextFeature för att hämta cursorns nästa feature  
Set pFeature = pFCursor.NextFeature  
Loop
```

End Sub

Bilaga 4 DelaUppLank

Option Explicit

```
Public Sub Noder1()
```

```
'****
```

```
Koden väljer värdena på vänster sida om bindestrecket ("-") ur kolumnen GrpLenke och sparar dem i en ny kolumn; Noder1.
```

```
'****
```

```
'Deklarerar och fastställer en variabel för kartdokumentet  
Dim pMxDoc As IMxDocument  
Set pMxDoc = ThisDocument
```

```
'Deklarerar en IFeatureLayer-variabel och fastställer den med  
'ContextItem-egenskap  
Dim pFLayer As IFeatureLayer  
Set pFLayer = pMxDoc.ContextItem
```

```
'Deklarerar och fastställer en variabel för lagrets feature-  
'klass  
Dim pFclass As IFeatureClass  
Set pFclass = pFLayer.FeatureClass
```

```
'Deklarerar och fastställer variabler för Fields-kollektionen  
Dim pFields As IFields  
Set pFields = pFclass.Fields
```

```
'Deklarerar och fastställer en heltalsvariabel som ska bevara  
'SummaTS-  
'fältets _indexpositionsnummer  
Dim intPosNoder1 As Integer  
intPosNoder1 = pFields.FindField("Noder_1")
```

```
'Deklarerar och fastställer variabel för att skapa nytt fält  
Dim pFieldEdit As IFieldEdit  
Set pFieldEdit = New Field  
Dim StrMedd1 As String
```

```
'Påbörjar If...Then för att se om fältet redan finns  
If intPosNoder1 = 1 Then  
    StrMedd1 = MsgBox("Fältet 'Noder_1' finns redan!",  
                    vbExclamation)  
Exit Sub
```

```
Else
```

```
'Ange egenskaper för fältets namn och typ  
pFieldEdit.Name = "Noder_1"  
pFieldEdit.Type = esriFieldTypeInteger
```

```
'Använd AddFieldmetod för att lägga till fält till tabellen  
pFclass.AddField pFieldEdit
```

```
'Återställer variabeln som innehåller Noder1-fältets index-  
positionsnummer  
intPosNoder1 = pFields.FindField("Noder_1")
```

```

End If

'Beräknar field-värden genom användandet av en cursor (markör)
'Uppdaterar markören med alla rader
Dim pCursor As ICursor
Set pCursor = pFclass.Update(Nothing, True)

'Definierar en kalkylator
Dim pCalc As ICalculator
Set pCalc = New Calculator

Set pCalc.Cursor = pCursor
With pCalc

    Set .Cursor = pCursor
    PreExpression = "Dim s As String" & vbNewLine & _
        "Dim sParts() As String" & vbNewLine & _
        "Dim t As String" & vbNewLine & _
        "Dim n As Integer" & vbNewLine & _
        "Dim i As Integer" & vbNewLine & _
        "Dim ipos As Integer" & vbNewLine & _
        "Dim Noder1 as Double" & vbNewLine & _
        "s = [GrpLenke]" & vbNewLine & _
        "sParts = Split(s, "" - "")" & vbNewLine & _
        "n = UBound(sParts)" & vbNewLine & _
        "t = sParts(n - 1)" & vbNewLine & _
        "For i = 1 To Len(t)" & vbNewLine & _
        "If IsNumeric(Left(t, i)) Then ipos = i" & _
        vbNewLine & _
        "Next" & vbNewLine & _
        "Noder1 = CLng(Left(t, ipos))"

    .Expression = "Noder1"
    .Field = "Noder_1"
End With
'Beräknar fältets värden
pCalc.Calculate

Set pFields = Nothing
Set pFclass = Nothing

```

End Sub

Public Sub Noder2()

'****

Koden väljer värdena på höger sida om bindestrecket ("-") ur kolumnen GrpLenke och sparar dem i en ny kolumn; Noder2.

'****

```

'Deklarerar och fastställer en variabel för kartdokumentet
Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument

```

```

'Deklarerar en IFeatureLayer-variabel och fastställer den med
'ContextItem-egenskap
Dim pFLayer As IFeatureLayer

```

```

Set pFLayer = pMxDoc.ContextItem

'Deklarerar och fastställer en variabel för lagrets feature-
'klass
Dim pFclass As IFeatureClass
Set pFclass = pFLayer.FeatureClass

'Deklarerar och fastställer variabler för Fields-kollektionen
Dim pFields As IFields
Set pFields = pFclass.Fields
'Deklarerar och fastställer en heltalsvariabel som ska bevara
SummaTS-
'fältets indexpositionsnummer
Dim intPosNoder2 As Integer
intPosNoder2 = pFields.FindField("Noder_2")

'Deklarerar och fastställer variabel för att skapa nytt fält
Dim pFieldEdit As IFieldEdit
Set pFieldEdit = New Field
Dim StrMedd2 As String

'Påbörjar If Then för att se om fältet redan finns
If intPosNoder2 = 1 Then
    StrMedd2 = MsgBox("Fältet 'Noder_2' finns redan!",
        vbExclamation)
    Exit Sub
Else
    'Ange egenskaper för fältets namn och typ
    pFieldEdit.Name = "Noder_2"
    pFieldEdit.Type = esriFieldTypeInteger

    'Använd AddFieldmetod för att lägga till fält till tabellen
    pFclass.AddField pFieldEdit

    'Återställer variabeln som innehåller Noder1-fältets index-
    positionsnummer
    intPosNoder2 = pFields.FindField("Noder_2")
End If

'Beräknar field-värden genom användandet av en cursor (markör)
'Uppdaterar markören med alla rader
Dim pCursor As ICursor
Set pCursor = pFclass.Update(Nothing, True)

'Definierar en kalkylator
Dim pCalc As ICalculator
Set pCalc = New Calculator

Set pCalc.Cursor = pCursor
with pCalc

    Set .Cursor = pCursor
    PreExpression = "Dim s As String" & vbNewLine & _
        "Dim sParts() As String" & vbNewLine & _
        "Dim t As String" & vbNewLine & _
        "Dim n As Integer" & vbNewLine & _
        "Dim i As Integer" & vbNewLine & _
        "Dim ipos As Integer" & vbNewLine & _

```

```
        "Dim Noder2 as Double" & vbNewLine & _
        "s = [GrpLenke]" & vbNewLine & _
        "sParts = Split(s, "" - "")" & vbNewLine & _
        "n = UBound(sParts)" & vbNewLine & _
        "t = sParts(n)" & vbNewLine & _
        "For i = 1 To Len(t)" & vbNewLine & _
        "If IsNumeric(Left(t, i)) Then ipos = i" &
        vbNewLine & _
        "Next" & vbNewLine & _
        "Noder2 = CLng(Left(t, ipos))"

        .Expression = "Noder2"
        .Field = "Noder_2"
    End With
    'Beräknar fältets värden
    pCalc.Calculate

    Set pFields = Nothing
    Set pFclass = Nothing
```

End Sub

Bilaga 5 TotLanksumma

Option Explicit

```
Public Sub BeraknaTotLanksumma()
```

```
'****
```

Koden jämför parvis värdena ur kolumnerna Noder1 och Noder2 med värden ur samma kolumner för att hitta matchande par. Om par hittas, både likadana och spegelvända, kommer motsvarande värden i radernas SummaTS-kolumn att summeras och läggas till i en ny kolumn; Lanksumma.

```
'****
```

```
'Deklarerar och fastställer en variabel för kartdokumentet
```

```
Dim pMxDoc As IMxDocument
```

```
Set pMxDoc = ThisDocument
```

```
'Deklarerar en IFeatureLayer-variabel och fastställer den med  
'ContextItem-egenskap
```

```
Dim pFLayer As IFeatureLayer
```

```
Set pFLayer = pMxDoc.ContextItem
```

```
Dim pFClass As IFeatureClass
```

```
Set pFClass = pFLayer.FeatureClass
```

```
'Deklarerar och fastställer variabler för att skapa feature  
'cursors
```

```
Dim pFCursor1 As IFeatureCursor
```

```
Dim pFCursor2 As IFeatureCursor
```

```
Dim pFCursor3 As IFeatureCursor
```

```
'Deklarerar IFeature-variabler
```

```
Dim pFeature1 As IFeature
```

```
Dim pFeature2 As IFeature
```

```
Dim pFeature3 As IFeature
```

```
'Deklarerar och fastställer heltalsvariabler som ska bevara  
'Noder_1, Noder_2, Summa_TS och Lanksumma fältens  
'indexpositionsnummer
```

```
Dim intPosNoder1 As Integer
```

```
intPosNoder1 = pFClass.FindField("Noder_1")
```

```
Dim intPosNoder2 As Integer
```

```
intPosNoder2 = pFClass.FindField("Noder_2")
```

```
Dim intPosSummaTS As Integer
```

```
intPosSummaTS = pFClass.FindField("Summa_TS")
```

```
Dim intPosLanksumma As Integer
```

```
intPosLanksumma = pFClass.FindField("Lanksumma")
```

```
'Deklarerar och fastställer variabler för Fields-kollektionen
```

```
Dim pFields As IFields
```

```
Set pFields = pFClass.Fields
```

```
'Påbörjar If Then för att se om fältet Lanksumma redan finns
```

```

Dim StrMedd As String
If intPosLanksumma = 1 Then
    StrMedd = MsgBox("Fältet 'Lanksumma' finns redan!",
                    vbExclamation)
    Exit Sub
Else
    'Deklarerar och fastställer variabel för att skapa nytt fält
    Dim pFieldEdit As IFieldEdit
    Set pFieldEdit = New Field

    'Ange egenskaper för fältets namn och typ
    pFieldEdit.Name = "Lanksumma"
    pFieldEdit.Type = esriFieldTypeInteger

    'Använd AddFieldmetod för att lägga till fält till tabellen
    pFclass.AddField pFieldEdit

    'Återställer variabeln som innehåller SummaTS-fältets index-
    'positionsnummer
    intPosLanksumma = pFields.FindField("Lanksumma")
End If

'Set till att nödvändiga fält finns i lagret
If intPosNoder1 = -1 Or intPosNoder2 = -1 Or intPosSummaTS = -1
Then
    MsgBox "De korrekta fälten finns ej i angivet lager"
    Exit Sub
End If

'Uppdaterar markörerna med alla rader
Set pFCursor1 = pFclass.Search(Nothing, False)
Set pFCursor2 = pFclass.Search(Nothing, False)

'Fastställer IFeature-variabeln med cursorns NextFeaturemetod
Set pFeature1 = pFCursor1.NextFeature

'Deklarerar två filter och variablerna som ska hålla värdena i
'looparna
Dim pQf As IQueryFilter
Dim pQf2 As IQueryFilter
Dim xVal1 As Integer
Dim xVal2 As Integer

Dim Medd As String
Medd = MsgBox("Beräkningen kommer nu att påbörjas." & vbNewLine
            & _"Tabellfönstret öppnas när beräkningen är klar."
            & vbNewLine & "Det kommer att ta någon minut.", _
            vbOKOnly + vbInformation + vbDefaultButton2,
            Beräkning av länksummor")

'Loopar genom alla features för att hitta de värden som matchar
'indexnr.
Do Until pFeature1 Is Nothing
    'Skapar ett queryfilter för att se om några indexnr i
    'feature1
    'matchar dem i feature2
    xVal1 = pFeature1.Value(intPosNoder1)
    xVal2 = pFeature1.Value(intPosNoder2)

```

```

Set pQf = New QueryFilter
pQf.WhereClause = "(Noder_1 = " & xVal2 & " And Noder_2 = "
                  & xVal1 & ") Or " + "(Noder_1 = " & xVal1
                  & " And Noder_2 = " & xVal2 & ")"

'Cursor1 letar genom hela tabellen efter värden som matchar
'WC ovan.
Set pFCursor2 = pFclass.Search(pQf, False)
If pFclass.FeatureCount(pQf) > 0 Then
    Set pFeature2 = pFCursor2.NextFeature
    Dim Sum As Integer
    Sum = 0

    'Om matchande värden finns så summeras deras SummaTS för
    'motsvarande indexnr i variabeln Sum tills inga fler
    'matchningar finns
    Do Until pFeature2 Is Nothing
        Sum = Sum + _
            pFeature2.Value(intPosSummaTS)
        Set pFeature2 = pFCursor2.NextFeature
    Loop

    'Om matchande värden finns så ska den totala Sum läggas
    'till i motsvarande indexnr. för "Lanksumma"
    Set pQf2 = New QueryFilter
    pQf2.WhereClause = "(Noder_1 = " & xVal2 & " And Noder_2
                       = " & xVal1 & ") Or " + _
                       "(Noder_1 = " & xVal1 & " And Noder_2
                       = " & xVal2 & ")"

    'Cursor1 letar genom hela tabellen efter värden som
    'matchar ovan.
    Set pFCursor3 = pFclass.Search(pQf2, False)
    If pFclass.FeatureCount(pQf2) > 0 Then
        Set pFeature3 = pFCursor3.NextFeature

        'Lägger till Sum i "Lanksumma" tills inga fler
        'matchande indexnr. finns

        Do Until pFeature3 Is Nothing
            pFeature3.Value(intPosLanksumma) = Sum
            pFeature3.Store
            Set pFeature3 = pFCursor3.NextFeature
        Loop
    End If
End If
Set pFeature1 = pFCursor1.NextFeature
Loop

Call OppnaTabellen.Tabellen

```

End

Sub

Bilaga 6 OppnaTabellen

Option Explicit

```
Public Sub Tabellen()
```

```
'****
```

```
Öppnar attributtabellen.
```

```
'****
```

```
'Get the selected item from the current contents view
Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument

Dim pUnknown As IUnknown
Set pUnknown = pMxDoc.SelectedItem

Dim pTablewindow2 As ITablewindow2
Set pTablewindow2 = New Tablewindow

Dim pExistingTablewindow As ITablewindow
Dim SetProperty As Boolean

Dim pLayer As ILayer
Dim pStandaloneTable As IStandaloneTable

' Bestäm det valda objektets typ
' Gå ut ur suben om objektet inte är ett featurelager eller en
'fristående tabell
If TypeOf pUnknown Is IFeatureLayer Then 'Ett FeatureLayer
    Set pLayer = pUnknown
    Set pExistingTablewindow = _
    pTablewindow2.FindViaLayer(pLayer)

    ' Kontrollera om en tabell redan finns; om inte så skapa en
    If pExistingTablewindow Is Nothing Then
        Set pTablewindow2.Layer = pLayer
        SetProperty = True
    End If

ElseIf TypeOf pUnknown Is IStandaloneTable Then
    ' En fristående tabell
    Set pStandaloneTable = pUnknown
    Set pExistingTablewindow = _
    pTablewindow2.FindViaStandaloneTable(pStandaloneTable)
    ' Kontrollera om en tabell redan finns; om inte så skapa en

    If pExistingTablewindow Is Nothing Then
        Set pTablewindow2.StandaloneTable = pStandaloneTable
        SetProperty = True
    End If
End If

If SetProperty Then
    pTablewindow2.TableSelectionAction = esriSelectFeatures
    pTablewindow2.ShowSelected = False
    pTablewindow2.ShowAliasNamesInColumnHeadings = True
```



```
    Set pTablewindow2.Application = Application  
Else  
    Set pTablewindow2 = pExistingTablewindow  
End If
```

```
'säkerställ att tabellen är synlig  
If Not pTablewindow2.IsVisible Then pTablewindow2.Show True
```

End Sub

Bilaga 7 RaderaFields

Option Explicit

```
Public Sub RaderaField()
```

```
'****
```

```
Raderar kolumnerna Noder1, Noder2, SummaTS och Lanksumma.
```

```
'****
```

```
'Deklarerar och fastställer en variabel för kartdokumentet  
Dim pMxDoc As IMxDocument  
Set pMxDoc = ThisDocument
```

```
'Deklarerar en IFeatureLayer-variabel och fastställer den med  
'ContextItem-egenskap  
Dim pFLayer As IFeatureLayer  
Set pFLayer = pMxDoc.ContextItem
```

```
'Deklarerar och fastställer en variabel för lagrets feature-  
'klass  
Dim pFclass As IFeatureClass  
Set pFclass = pFLayer.FeatureClass
```

```
'Deklarerar och fastställer variabler för  
'Fields(kolumn)kollektionen  
Dim pFields As IFields  
Set pFields = pFclass.Fields
```

```
'Deklarerar och fastställer en heltalsvariabel som ska bevara  
'SummaTS-fältets indexpositionsnummer  
Dim intPosNoder1 As Integer  
intPosNoder1 = pFields.FindField("Noder_1")
```

```
Dim pField1 As IField  
Set pField1 = pFields.Field(intPosNoder1)
```

```
If intPosNoder1 = -1 Then  
Exit Sub  
Else  
'Raderar namngivet fält  
pFclass.DeleteField pField1  
End If
```

```
'Deklarerar och fastställer en heltalsvariabel som ska bevara  
'SummaTS-fältets indexpositionsnummer  
Dim intPosNoder2 As Integer  
intPosNoder2 = pFields.FindField("Noder_2")
```

```
Dim pField2 As IField  
Set pField2 = pFields.Field(intPosNoder2)
```

```
If intPosNoder2 = -1 Then  
Exit Sub  
Else  
'Raderar namngivet fält  
pFclass.DeleteField pField2
```

End If

'Deklarerar och fastställer en heltalsvariabel som ska bevara
'SummaTS-fältets indexpositionsnummer

```
Dim intPosSummaTS As Integer  
intPosSummaTS = pFields.FindField("Summa_TS")  
Dim pFieldSummaTS As IField  
Set pFieldSummaTS = pFields.Field(intPosSummaTS)
```

```
If intPosNoder1 = -1 Then  
    Exit Sub  
Else
```

```
'Raderar namngivet fält  
pFclass.DeleteField pFieldSummaTS  
End If
```

'Deklarerar och fastställer en heltalsvariabel som ska bevara
'SummaTS-fältets indexpositionsnummer

```
Dim intPosLanksumma As Integer  
intPosLanksumma = pFields.FindField("Lanksumma")
```

```
Dim pFieldLanksumma As IField  
Set pFieldLanksumma = pFields.Field(intPosLanksumma)
```

```
If intPosNoder1 = -1 Then  
    Exit Sub
```

```
Else  
'Raderar namngivet fält  
pFclass.DeleteField pFieldLanksumma  
End If
```

MsgBox "Fälten är raderade"

End Sub

Bilaga 8 ThisDocument

Option Explicit

```
Private Sub Radera_Click()
```

```
'****
```

```
Anropar proceduren RaderaFields
```

```
'****
```

```
    Call RaderaFields.RaderaField
```

```
End Sub
```

```
Private Sub SkapaNyaPunkter_MouseDown(ByVal button As Long, ByVal  
    shift As Long, ByVal x As Long, ByVal y As Long)
```

```
'****
```

```
Verktyg som skapar nya punkter där användaren klickar i  
användarfönstret.
```

```
'****
```

```
    Dim pMap As IMap
```

```
    Dim pMxDoc As IMxDocument
```

```
    Set pMxDoc = ThisDocument
```

```
    Set pMap = pMxDoc.FocusMap
```

```
    'Get the first layer in the map
```

```
    Dim pFeatLyr As IGeoFeatureLayer
```

```
    Set pFeatLyr = pMap.Layer(0)
```

```
    Dim tmpPoint As IPoint
```

```
    Set tmpPoint = New Point
```

```
    Set tmpPoint=pMxDoc.ActiveView.ScreenDisplay.  
        DisplayTransformation.ToMapPoint(x, y)
```

```
    'Get the feature class
```

```
    Dim pFclass As IFeatureClass
```

```
    Set pFclass = pFeatLyr.FeatureClass
```

```
    'Create the new point feature
```

```
    Dim pFeat As IFeature
```

```
    Set pFeat = pFclass.CreateFeature
```

```
    Set pFeat.Shape = tmpPoint
```

```
    Dim pDataset As IDataset
```

```
    Set pDataset = pFclass
```

```
    Dim pworkspace As IWorkspace
```

```
    Set pworkspace = pDataset.Workspace
```

```
    Dim pFeatureworkspace As IFeatureworkspace
```

```
    Set pFeatureworkspace = pworkspace
```

```
    Dim pFeatureClass As IFeatureClass
```

```
    Set pFeatureClass = pFeatureworkspace.OpenFeatureClass("Flis")
```

'HÄR VILL VI HA IN ANVÄNDARENS VAL

pFeat.Store

'Refresh the view
pMxDoc.ActiveView.Refresh

End Sub

Private Sub SkapaNyaLinjer_MouseDown(ByVal button As Long, ByVal shift As Long, ByVal x As Long, ByVal y As Long)

'****

Verktyg som skapar nya linjer där användaren klickar i användarfönstret. Ett dubbelklick avslutar figuren.

'****

Dim pMxDoc As IMxDocument
Dim pLine As IPolyline
Dim pRubberLine As IRubberBand

'QI for the MXDocument interface
Set pMxDoc = ThisDocument

Dim pMap As IMap
Set pMap = pMxDoc.FocusMap

'Get the first layer in the map
Dim pFeatLyr As IGeoFeatureLayer
Set pFeatLyr = pMap.Layer(0)

Dim pFclass As IFeatureClass
Set pFclass = pFeatLyr.FeatureClass

Set pLine = New Polyline
Dim pFeat As IFeature
Set pFeat = pFclass.CreateFeature
Dim pGraCont As IGraphicsContainer

'QI for the IGraphicsContainersSelect interface on the document's
'activeview
Set pGraCont = pMxDoc.ActiveView

'Create a new RubberLine
Set pRubberLine = New RubberLine

'Check which mouse button was pressed...
If button = 1 Then ' If button 1 (left) then create a new
Line
(TrackNew)

'Return a new Line from the tracker object using
'TrackNew
Set pLine = pRubberLine.TrackNew
(pMxDoc.ActiveView.ScreenDisplay, Nothing)

Set pFeat.Shape = pLine
pFeat.Store

```
        'pMxDoc.ActivatedView.Refresh
    Else 'If button 2 (right) then display a message
        MsgBox " Use the Left Button for Digitizing! "
    End If

    'Refresh the activeView
    pMxDoc.ActiveView.Refresh
```

End Sub

```
Private Sub SkapaNyttLager_Click()
```

```
    '****
    Visar formuläret NyttLager.
    '****
```

```
        frmNyttLager.Show
```

End Sub

```
Private Sub NyTabell_Click()
```

```
    '****
    Anropar Ny; en tom tabell.
    '****
```

```
        Call NyaTabellen.Ny
```

End Sub

```
Private Sub BeraknaLankar_Click()
```

```
    '****
    Anropar procedurerna SummaTS, Noder1, Noder2 och
    BeraknaTotLanksumma.
    '****
```

```
        Call TSSumman.SummaTS
        Call DelaUppLank.Noder1
        Call DelaUppLank.Noder2
        Call TotLanksumma.BeraknaTotLanksumma
```

End Sub

Bilaga 9 NyaTabellen

Option Explicit

Sub Ny()

'****

Namnger en ny tabell, anger var den skall sparas, samt lägger till tabellen i innehållsförteckningen.

'****

```
Dim pTab As ITable
'Ange tabellens namn och var den ska sparas nedan
Set pTab = CreateDbf("Flöden", "C:\ArcObjects")
AddTableToTOC pTab
MsgBox "Den nya tabellen har lagts till i TOC."
```

End Sub

Public Function CreateDbf(strName As String, strFolder As String) As ITable

'****

Skapar en tom fristående tabell. Användaren kan ange hur många kolumner som önskas samt namnge dem.

'****

```
'Funktionen skapar en dbf
On Error GoTo EH

'Öppna "arbetsrummet"
Dim pFWS As IFeatureWorkspace
Dim pworkspaceFactory As IworkspaceFactory
Dim fs As Object
Dim pFieldsEdit As IFieldsEdit
Dim pFieldEdit As IFieldEdit
Dim pField As IField
Dim pFields As IFields

Set pworkspaceFactory = New ShapefileworkspaceFactory
Set fs = CreateObject("Scripting.FileSystemObject")

If Not fs.FolderExists(strFolder) Then
    MsgBox "Folder does not exist: " & vbCr & strFolder
    Exit Function
End If

Set pFWS = pworkspaceFactory.OpenFromFile(strFolder, 0)

'Skapar fälten som används av objektet
Set pFields = New Fields
Set pFieldsEdit = pFields
pFieldsEdit.FieldCount = 3

'Tabellens första fält/kolumn
Set pField = New Field
Set pFieldEdit = pField
```

```

With pFieldEdit
    .Length = 6 'namnets längd
    .Name = "Länk"
    .Type = esriFieldTypeInteger
End with

Set pFieldsEdit.Field(0) = pField
Set pField = New Field
Set pFieldEdit = pField

With pFieldEdit
    .Length = 6
    .Name = "Flöde"
    .Type = esriFieldTypeDouble
End with

Set pFieldsEdit.Field(1) = pField
Set pField = New Field
Set pFieldEdit = pField

With pFieldEdit
    .Length = 6
    .Name = "?"
    .Type = esriFieldTypeDouble
End with

Set pFieldsEdit.Field(2) = pField

Set Createdbfc = pFWS.CreateTable(strName, pFields, Nothing,
    Nothing, "")
Set pFWS = Nothing
Set pWorkspaceFactory = Nothing
Set fs = Nothing
Set pFieldsEdit = Nothing
Set pFieldEdit = Nothing
Set pField = Nothing

Exit Function
EH:
    MsgBox Err.Description, vbInformation, "createdbfc"

```

End Function

```

Private Sub AddTableToTOC(pTable As ITable)

'****
Lägger till tabellen till innehållsförteckningen.
'****

'Procedur för att lägga till tabellen till kartans
'innehållsförteckning (TOC)
    Dim pDoc As IMxDocument
    Dim pMap As IMap
    Dim intCol As Integer
    Dim blnExists As Boolean
    Set pDoc = ThisDocument
    Set pMap = pDoc.FocusMap

```



```
'Skapar en nytt fristående tabell och lägger till den till
'fokuskartans kollektion
Dim pStTab As IStandaloneTable
Set pStTab = New StandaloneTable
Set pStTab.Table = pTable
Dim pStTabColl As IStandaloneTableCollection
Set pStTabColl = pMap

'Innan tabellen läggs till, kollar om den redan finns
For intCol = 0 To pStTabColl.StandaloneTableCount - 1
    If UCase(pStTabColl.StandaloneTable(intCol).Name) =
        UCase(pStTab.Name) Then
        blnExists = True
        Exit For
    End If
Next

'MsgBox " Tabellen skapas ", vbInformation

If blnExists = False Then
    pStTabColl.AddStandaloneTable pStTab
End If

'Uppdaterar TOC
pDoc.UpdateContents
pDoc.ActiveView.Refresh

'Töm alla variabler på minnen
Cleanup:
Set pDoc = Nothing
Set pMap = Nothing
Set pStTab = Nothing
Set pStTabColl = Nothing

End Sub
```
