



Térinformatikai fejlesztések hálózat topológiák alkalmazásával  
doktori (PhD) értekezés

GIS Developments Applying Network Topologies  
Thesis of PhD Dissertation

Zichar Marianna

Témavezető: Dr. habil. Szabó József

Debreceni Egyetem  
Debrecen, 2003.



# 1. Bevezetés

A vektor alapú térinformációs rendszerek közül a topológiai adatmodellt használók alkalmasak a legváltozatosabb adatelemzések végrehajtására. Ezen rendszerek explicit formában tartalmazzak szomszédsági információkat, melyek segítségével három típusú topológia definiálható. Eredményeim a hálózat típusú topológiák elemző műveleteire vonatkoznak, melyet egy Visual LISP nyelven kifejlesztett Autodesk Map alkalmazás szemléltet.

Az Autodesk Map egy AutoCAD alapú specializált térképezési szoftvertermék, amely tartalmazza az AutoCAD program teljes funkcionalitását kiegészítve hatékony, kifejezetten térképészeti és térinformatikai szakemberek részére készített eszközökkel. Az eredmények gyakorlati bemutatásához igénybe vettem az Autodesk Map testreszabási lehetőségeit is, saját legördülő menüt, párbeszédpaneleket, és eszköztárat is definiáltam.

Az alkalmazás felhasználja a *pont a poligonban* algoritmus térinformatikai változatának általam általánosított formáját is.

Az Autodesk Map használata során általam és kollégáim által hiányolt néhány új parancsot megalkottam, melyek akár AutoCAD környezetben is jól használhatóak.

## 2. Városi tömegközlekedés modellezése hálózat topológiával

A hálózat topológia kiválóan alkalmas egy település utcáinak egységbe foglalására elemző műveletek végrehajtásának céljából. Egyik legismertebb művelet az optimális útvonal meghatározása, mely egy kiindulási pontból induló és oda visszaérkező élek olyan sorozatát határozza meg, mely előre meghatározott csomópontokat is érint. Ha a csomópontok megközelítéséhez tömegközlekedési eszközök használatát is engedélyezzük, akkor ezek nyomvonalával bővítenünk kell a topológiát. Ennek eredményeképpen az elemző műveletek hatáskörébe mindig beletartozik a topológiát alkotó összes objektum.

Célom az volt, hogy a célpontok megközelítéséhez felhasználható tömegközlekedési eszköz típusainak megválasztása tetszőlegesen szabályozható legyen, azaz a hálózat topológia, a felhasználó beállításaitól függő *résztopológiáján* is működjenek az elemző függvények.

## 2.1. Rézstopológia fogalma

**1. Definíció.** Tekintsük pontok  $N$ , és élek  $L$  halmazát a következőképpen:

$$N = \{p_i \mid i = 1..m\} \text{ és } N \neq \emptyset$$
$$L = \{l \mid l = (p_j, p_k) \text{ ahol } j, k \leq m\} \text{ és } L \neq \emptyset$$

Az így kialakított objektumok halmazát  $H(N, P)$ -vel jelöljük és a továbbiakban **hálózatnak** nevezzük.

**2. Definíció.** Tekintsük a  $H_1(N_1, P_1)$  és a  $H_2(N_2, P_2)$  hálózatot, melyekre teljesül  $N_1 = \{p_i \mid i = 1..m_1\}$   $N_2 = \{q_i \mid i = 1..m_2, m_1 \geq m_2\}$   $N_1 \cap N_2 = \emptyset$   $L_1 \cap L_2 = \emptyset$

A  $H(N, L)$  hálózatot a  **$H_1$  hálózat  $H_2$ -vel való bővítésének** nevezzük és  **$H = H_1 + H_2$**  módon jelöljük, ha

$$N = N_1 \cup N_2$$
$$L = L_1 \cup L_2 \cup L_3, L_3 \neq \emptyset$$
$$L_3 = \{l \mid l = (p_j, q_k) \text{ ahol } p_j \in H_1, q_k \in H_2, j = 1..m, m \leq m_2\}$$

összefüggések érvényesek.

**Jelölés:** Egy  $H(N, P)$  hálózat objektumain definiált topológiát jelöljük a továbbiakban  $T(H)$ -val.

**3. Definíció.** A  $T(H)$  topológiát **helyesnek** nevezzük, ha topológiai adatai konzisztensek, azaz nem tartalmaznak érvénytelen hivatkozásokat.

Az adatok konzisztenciáját megszüntetheti például egy olyan csomópont törlése, mely még részt vett él objektum meghatározásában. Ebben az esetben a csomópont nélkül maradt él objektumok törlésével, vagy a hiányzó csomópont pótlásával az adatok konzisztenciája helyreállítható.

**4. Definíció.** A  $T(H)$ ,  $H = H_1 + H_2$  topológia  **$T(H) - H_2$ -vel jelzett résztopológiáján** azt a topológiát értjük, mely a  $T(H)$  topológia  $H_2$ -beli éleinek eltávolításával és a keletkezett izolált csomópontok megszüntetésével áll elő.

**1. Állítás.** A  $T(H)$ ,  $H = H_1 + H_2$  topológia  $T(H) - H_2$  résztopológiája helyes topológia.

**Bizonyítás:**

Élek törlése nem befolyásolja a topológia helyességét, mert a szomszédsági információk a kezdő és végpontot definiáló csomópontok azonosítójával kerülnek tárolásra.

A csomópontok esetében a definíció szerint csak azok szűnnek meg, melyekre már nem illeszkedik él, tehát nincs már az adatok közt olyan él objektum, mely az eltávolított csomópontokra hivatkozna.

**Következmény:** *A  $T(H)$ - $H_2$  résztopológia alkalmas elemző műveletek végrehajtására.*

A 2. Definíció általánosításával egy hálózat többszörösen való bővítése határozható meg. A többszörösen bővített hálózaton definiált topológiából számtalan résztopológia keletkezhet a 4. Definícióban meghatározott módon. Az ilyenformán létrejött résztopológiákra teljesül az 1.Állítás alábbi, általánosított változata:

**2. Állítás.** *A  $T(H)$ ,  $H=H_1+H_2+\dots+H_n$  topológia  $T(H)$ - $H_i$  résztopológiája helyes topológia minden  $i=1..n$  értékre.*

**Bizonyítás:**

A bizonyítás menetét nem befolyásolja, hogy a H hálózat többszörös bővítéssel keletkezett. Az állítás bizonyítása megegyezik az előző bizonyítás gondolatmenetével.

## 2.2. Résztopológián történő elemzés megvalósítása

Az elméleti modellt alkotó entitások körének, és az entitások attribútumainak meghatározását követően vonalláncok alkalmazásával építettem ki az adatmodellt Autodesk Map 6 környezetben. A szükséges attribútumok az utcanevek kivételével objektumadat táblában tárolódnak. Az utcanevek tárolására külső adatbázist (Excel táblát) alkalmaztam, mert így a szükséges karbantartási, frissítési műveletek lényegesen könnyebben elvégezhetőek.

A célként kitűzött leggyorsabb útvonal meghatározásához az összes vonalláncként definiált, a közlekedéssel kapcsolatos fólián található objektumból egyetlen hálózat topológiát definiáltam. A létrejött topológiai adatok közül a tömegközlekedési eszközök nyomvonalát reprezentáló vonalláncok bejárési irányát meghatározó *Direction* mező értékének értelemszerű megváltoztatásával „egyirányúsítani” kellett ezen éleken történő haladást. Ehhez, illetve az objektumadatok rajzelemekhez történő csatolásához definiáltam egy függvényt

*(feltolt folia\_nev tabla mezo rsz ertek)*

formában. Ennek segítségével a paraméterben megadott fólián található objektumok megadott táblához tartozó rekordjának mezőjének adható meg (új) érték. Ha valamelyik objektumhoz még egyáltalán nem, vagy nem elegendő

számú rekord tartozik, akkor előbb új rekordot csatol a függvény az érintett objektumhoz, majd feltölti a mező értékét.

A résztopológia gyakorlati megvalósításához az alapötletet a következő megfigyelés szolgáltatta. Ha a topológiát alkotó objektumok mind a forrásrajzban mind a projektrajzban jelen vannak, akkor a topológia memóriába való betöltésénél kiválasztható, hogy honnan kerüljenek az adatok a memóriába. A topológia alkalmazásból való betöltéséhez használható VLISP függvény paramétereivel szintén szabályozható az adatok eredete.

A topológia forrásrajzból történő betöltése esetén az objektumok köre nem szabályozható. Ha viszont meg tudjuk oldani, hogy a hálózatot alkotó objektumoknak csak egy olyan részhalmaza kerüljön bemásolásra a projektrajzba, melyből betöltött topológia állapotának lekérdezése során teljesnek és helyesnek bizonyul, úgy a projektrajz objektumain hálózat elemzés végezhető.

A forrásrajz erre a célra kialakított fólia szerkezete és az általam definiált

(*geometria\_betolt\_folia\_lista*)

VLISP függvény lehetővé teszi, hogy Tulajdonság lekérdezés definiálásával mindig a felhasználó beállításainak megfelelő résztopológiát alkotó objektumok kerüljenek a projektrajzba lehetővé téve ezzel az elemző műveletek sikeres használatát.

Az elemzés végeredményeként egy új, *ut* nevű hálózat topológia jön létre a projektrajzon, az út bejárásához szükséges információk pedig egy BR\_ut nevű objektumadat táblában kerülnek rögzítésre. Az elemző műveletet vezérlő profile állományban kérhető lenne ugyan az eredmény automatikus kiemelése, de jobbnak láttam egy saját függvény definiálását erre a célra. Az

(*utvonal\_megjelenitese*)

függvény különböző színek, és vonaltípusok használatával jeleníti meg az ajánlott útvonalat az Utvonal fólián. A rajzterületen megjelenő jelmagyarázat segítségével pedig könnyen nyomon követhetővé válik a felhasznált tömegközlekedési eszközök típusai.

Az alkalmazás hatékonyságának növeléséhez biztosítanom kellett, hogy az előállított információ valamilyen módon szabadon „elvihető” legyen. Ennek megvalósításához az alkalmazás menürendszerébe két további almenüt építettem be, melyekkel az elkészült útvonal ajánlatról térképrészlet, illetve szöveges riport nyomtatása kezdeményezhető. A riport megjeleníthető a képernyőn is egy párbeszédpanel segítségével, melyben elhelyezett nyomógombbal szintén kezdeményezhető annak kinyomtatása.

## 2.3. Az alkalmazás további funkciói; egyedi megoldások

Az útvonal során meglátogatni kívánt célok (általában épületeket) egy nagyobb csoportból kell kiválasztanunk, amit egy felhasználóbarát alkalmazásnak illik minél jobban elősegíteni. Ebből a célból a párbeszédpanel bármelyik listadobozában kijelölt elemről automatikusan megjelenik egy rövid szöveges ismertető, sőt egy nyomógombbal megtekinthetünk egy fotót is róla, megkönnyítve ezzel a felhasználó döntését.

A hálózatelemző műveletek használatához a felhasználó által a párbeszédpanelben kiválasztott célok topológiai csomópont azonosítójának gyors elérésére volt szükségem. Ehhez egy olyan szöveges állományt készítettem (*latniv\_hu.txt*), melynek minden sora egy-egy épületről tartalmazza a következő információkat, egyetlen (!) szóközzel elválasztva:

- épület 5 betűs azonosítója,
- 4 hosszban a csomópont azonosító,
- épület megnevezése (az állomány névben megjelölt nyelven)

A szöveges állományból kiolvasott adatokat egyetlen, strukturált listává alakítom, ahol a lista elemei a szöveges állomány egy-egy sorában tárolt adatokból keletkeznek, az értekezésben részletezett szabályok szerint. Minden elem tartalmaz egy numerikus értéket is, mely a párbeszédpanelt alkotó mozaikokhoz való tartozást írja le. Így e programváltozó és két további saját fejlesztésű függvény segítségével nyomon követhető az egyes elemek két lista doboz közötti mozgása. A szerkezet egyik nagy előnye, hogy a listák betű szerinti rendezettségéhez elegendő a szöveges állomány épület elnevezések szerinti rendezettségének biztosítása.

Az épületekhez vezető éleket úgy alakítottam ki, hogy azok végpontja az épületeket reprezentáló poligonok belsejébe vezessenek. Ezzel lehetővé tettem, hogy a *pont a poligonban* tesztet is felhasználó

(*latniv\_keszit*)

függvény alkalmazásával automatizálni lehessen a korábban említett szöveges állomány elkészítését, vagy a topológia újradefiniálásának esetén szükséges frissítését.

Egy utca térképen való pontos elhelyezkedésének meghatározásához egy felhasználóbarát funkcióval is kiegészítettem az alkalmazást. Egy listából kiválasztott – külső adatbázisban tárolt – utcanév kijelölését követően a neki megfelelő objektum kiemelésre kerül, és egyidejűleg a rajz nézete a láthatóságra optimalizálva kerül beállításra.

A CD mellékleten található Visual LISP nyelven kifejlesztett alkalmazás jelenleg két nyelven (magyar és angol) is tud kommunikálni a felhasználóval. A nyelv kiválasztásához külön eszköztárat definiáltam, mely segítségével bármikor megváltoztathatjuk az alkalmazás nyelvét. Az adatbázisok tervezésének szempontjából a nyelvfüggő adatokat (elnevezések, ismertetőik) több nyelven kell tárolni úgy, hogy elegendő egyszerre csak az alkalmazás aktuális nyelvén tárolt adatok elérhetőségét biztosítani.

Az ilyen problémák legkézenfekvőbb megoldása az adatok nyelvenként külön strukturált txt állományban való tárolása, melyből az aktuális nyelvre vonatkozó adatokat az alkalmazásból listába beolvasva, programváltozóként érjük el. Az értekezésben részletezett a útmutatást követve a használható nyelvek köre utólag is, tetszőlegesen tovább bővíthető, vagy akár szűkíthető anélkül, hogy az alkalmazás forráskódjában változtatásokra kényszerülnénk.

A modell és az alkalmazás kialakítása során egyaránt arra törekedtem, hogy a grafikus adatbázis utólag is lecserélhető legyen, nem feltétlen a fejlesztő által. Ennek köszönhetően, az értekezésben teljes terjedelmében ismertetett modell fizikai megvalósításához elegendő információ és útmutatás áll rendelkezésre, így bármelyik településről készített forrásrajz csatolható a projekthez.

### **3. A pont a poligonban teszt**

Számos térinformatikai alapismereteket tartalmazó könyvben megtalálhatjuk az algoritmus ismertetését, de sajnos gyakorta elnagyolva, és nem minden esetben jól működő változata lelhető fel. Az alábbi algoritmus alapötlete azonos az említettekkel, de egyéni megoldásokat is tartalmaz, mellyel egyúttal a felhasználhatósági körét is sikerült bővíteni.

Az alapprobléma a következő:

Adott egy poligon és egy pont, melyről el kell döntenünk, hogy a poligonon belül vagy kívül található.

A válasz meghatározását elősegítő alapötlet:

Számoljuk meg, hogy az adott pontból kiinduló, pozitív irányba mutató függőleges félegyenes hányszor metszi a poligon oldalait. Amennyiben ez a szám páros, vagy nulla, akkor a kérdéses pont külső pont, amennyiben páratlan, akkor pedig belső.

Az alapötlet nem rendelkezik viszont arról az esetről amikor a pontból indított félegyenesre illeszkedik a poligon valamelyik csúcsa. Az én elgondolásom szerint ilyen helyzetekben a csúcsponthoz tartozó poligon oldalak, a pontból kiinduló félegyeneshez való helyzete dönti el, hogy emelkedik-e a metszéspontok száma vagy sem. Eszerint csak abban az esetben

növeljük eggyel a számlálót, ha az említett oldalak a félegyenes különböző oldalán találhatóak. A poligon függőleges – tehát a félegyenessel párhuzamos – oldalait figyelmen kívül lehet hagyni az algoritmus során.

Az így pontosított algoritmus a legáltalánosabb esetekben is jól működik, akár konkáv poligonokról, nem egyszerű poligonokról (nem csak a szomszédos oldalaknak van közös pontjuk), vagy a szigetet tartalmazó poligonokról legyen szó. Ezek a speciálisnak mondható poligonok a gyakorlatban is előfordulhatnak elég csak az övezetekre (pufferekre) vagy törpeállamot tartalmazó országok térképére (Olaszország), esetleg nyitott, belső udvart tartalmazó épületek (pl. Református Kollégium) alaprajzára gondolnunk.

Az algoritmus Visual LISP nyelven implementált, és az alkalmazás által használt verziója az VInf.lsp állományban

(pont\_poligonban *p poligon*)

szintaktikával került definiálásra, ahol paraméterként a pont objektum koordinátáit, illetve a poligon rajzbeli azonosítóját kell megadnunk. A visszatérési érték értelemszerűen *nil*, ha külső pontot teszteltünk, és T ellenkező esetben.

## 4. Új AutoCAD parancsok

A szoftver használata, és az alkalmazás fejlesztése közben hiányolt szolgáltatásokat nem csak Visual LISP függvények, hanem – ahol volt értelme – AutoCAD parancsok formájában is elkészítettem. Az Autodesk Map az AutoCAD térinformatikai kiterjesztéseként jött létre, azaz tartalmazza az AutoCAD teljes funkcionalitását.

A kiválasztási halmaz tetszőleges – a definiálás módjától függő – objektumok, belső azonosítóval ellátott halmaza. Ha parancsok, vagy Visual LISP függvények eredményeként állt elő, akkor a rajz nézetét nem lehetett úgy módosítani, hogy a halmazt alkotó objektumok láthatóak legyenek. Az általam definiált új

(zoom\_kiv\_hz *kiv\_hz\_az*)

szintaktikájú Visual LISP függvény használatával alkalmazásokból megoldható vált a kiválasztási halmaz elemeire történő nagyítás.

A rajzterületen látható bekapcsolt fóliákon elhelyezkedő objektumok köre Autodesk Map-ben a projekt nézetétől függ, amit vagy manuálisan változtathatunk meg, vagy a forrásrajzok terjedelmével határozhatjuk meg. Ez utóbbi esetben a forrásrajz terjedelmének meghatározásában annak összes fóliája

automatikusan részt vesz, így nem valósítható meg a fóliaszintű nagyítás. E hiány pótlására definiáltam a

(*zoom\_folia\_folia\_lista*)

VLISP függvényt, mely a paraméterként megadott fóliák terjedelmét figyelembe véve állítja be a nézetet. A függvény AutoCAD parancs változatát

*zoom\_f*

formában hoztam létre, ahol a fólia lista a Fóliák választása párbeszédpanelben található listadoboz elemeinek kiválasztásával kerül meghatározásra.

Tetszőleges fólián, vagy fóliákon található összes objektum törlését egyetlen lépésben megvalósítja a

(*urit\_folia\_lista*)

szintaktikával definiált Visual LISP függvény, melynek parancs változata

*urit\_f*

formában érhető el. A függvény változat visszatérési értéke T, ha minden objektum sikeresen el lett távolítva a rajzból, ellenkező esetben *nil*. A parancs változathoz a fólia lista az előző parancsnál bemutatott párbeszédablak segítségével kerül meghatározásra.

A fejezetben bemutatott függvények és parancsok forráskódjai az értekezés CD mellékletén a *rutin.lsp* állományban megtalálhatóak. Az alkalmazás telepítésével ezen parancsok is betöltődnek és ezáltal használhatóvá válnak. Ha csak a parancsokat szeretnénk elérhetővé tenni, akkor elegendő a *rutin.lsp* állomány betöltése.

Végül kiemelem, hogy az inhomogén hálózat topológiából kialakított résztopológiákon történő elemzés megvalósítása jelentősen megnöveli az elemző műveletek hatékonyságát és felhasználhatósági körét, s egyben összetettebb vizsgálódások végrehajtását teszi lehetővé. Ezt a folyamatot könnyítik meg az újonnan definiált AutoCAD parancsok, melyek a dolgozat fő eredményétől függetlenül is jól alkalmazhatóak nem csak térinformatikai környezetben.

# 1. Introduction

The topologic data model is suitable for execution of the most varied data analysis from the vector based geographic information systems. These systems store information about the neighbourhood in an explicit form, which is able to define three types of topology. My results concern the analysing tools of network topology that are demonstrated by an application developed in programming language Visual LISP.

Autodesk Map is a GIS and mapping software that combines the power and precision of AutoCAD with the specialised tools for creating, maintaining and producing maps and geographic data. It is optimised for precision mapping and GIS analysis in the AutoCAD environment and used to solve both traditional and unique mapping challenges across many areas and industries. I used the tools of customisation in Autodesk Map to demonstrate the results in practice, so I defined custom popup menu, several dialogue boxes and a toolbar also.

The application applies the version of point in polygon test used in geographical information system, which was generalised by myself.

I developed some new command also, that my colleagues and I missed during working with Autodesk Map. They can be applied well in the environment of AutoCAD also.

## 2. The modelling of urban public transport by network topology

The network topology is rather suitable for creating a unit from the streets in a settlement in order to execute analysing tools. One of the most popular analysing tools is to determine the best route, which produces the series of links with the same starting and ending point, while visiting nodes given in advance. If the usage of taking public transport is allowed to reach the nodes, then the topology must be widened by the traces of public transport. Doing so the analysing tools use always the whole set of objects belonging to the topology.

My aim was that the type of public transport used to reach visit points could be selected without restriction. That means, the analysing tools would be able to use only a part-topology defined by the user input.

## 2.1. The concept of part-topology

**Definition 1.** Let us consider the set of points called  $N$  and the set of links called  $L$  in the following way:

$$N = \{p_i \mid i = 1..m\} \text{ and } N \neq \emptyset$$

$$L = \{l \mid l = (p_j, p_k) \text{ where } j, k \leq m\} \text{ és } L \neq \emptyset$$

Let us indicate the set of objects arisen this way with  $H(N, P)$  and call it network in what follows.

**Definition 2.** Let us consider the networks  $H_1(N_1, P_1)$  and  $H_2(N_2, P_2)$  satisfying  $N_1 = \{p_i \mid i = 1..m_1\}$   $N_2 = \{q_i \mid i = 1..m_2, m_1 \geq m_2\}$   $N_1 \cap N_2 = \emptyset$   $L_1 \cap L_2 = \emptyset$

The network  $H(N, L)$  is called **the enlargement of  $H_1$  with  $H_2$** , and is indicated with  $H = H_1 + H_2$  if it satisfies the following relations

$$N = N_1 \cup N_2$$

$$L = L_1 \cup L_2 \cup L_3, L_3 \neq \emptyset$$

$$L_3 = \{l \mid l = (p_j, q_k) \text{ where } p_j \in H_1, q_k \in H_2, j = 1..m, m \leq m_2\}.$$

**Notation.** Let us indicate the topology defined by the network  $H(N, P)$  with  $T(H)$  in what follows.

**Definition 3.** We say that the topology  $T(H)$  is correct, if the data of topology is consistent that is to say does not contain any invalid reference.

The consistence of data can be aborted by deleting a node, which is still determining a link. In this case the consistence can be restored by either deleting the link with one node, or replacing the missing node.

**Definition 4.** The part-topology denoted by  $T(H) - H_2$  of topology  $T(H)$ ,  $H = H_1 + H_2$  is the topology that is created by removing the links of  $T(H)$  belonging to  $H_2$ , and the appearing pseudo nodes.

**Theorem 1.** The part-topology  $T(H) - H_2$  of  $T(H)$ ,  $H = H_1 + H_2$  is correct.

**Proof.**

Deleting of nodes does not influence the correction of topology, because the information about neighbourhood is recorded by the identification numbers of starting and ending nodes.

Only nodes without connecting links are removed following the definition. This ensures us that no node exists in database, which would reference to the deleted node.

**Conclusion.** *The part-topology  $T(H)-H_2$  is suitable for execution of analysing tools.*

Generalising the Definition 2. the multiplied enlargement of network can be determined. Several part-topology can be created from the topology defined by a multiplied enlarged network applying the Definition 4. These part-topologies satisfy the following form of generalised Theorem 1.

**Theorem 2.** *Let us consider the topology  $T(H)$ , where  $H=H_1+H_2+...+H_n$ . All of its part-topology  $T(H)-H_i$  is correct for any values of  $i=1..n$ .*

**Proof.**

The order of proofs is not influenced by the fact, that the network is created by multiplied enlargement. The proof of the two theorems is identical

## 2.2. The realising of analysis of part-topology

After determining the entity types building the theoretical model, and their attributes I built the data model using polylines in the environment of Autodesk Map Release 6. The necessary attributes, except of the street names are stored in object data tables. To record the names of streets I used an external database (Excel table) in order to make easier its modification and updating.

A single network topology is defined from the all objects of polyline situated on the layers related to transport in order to determine the fastest route which was set as an aim. The established topologic data must be corrected, because the links belonging to public transport can not be bidirectional. To do so, the value of field called *Direction*, belonging to the polylines representing the trace of public transport must be altered. I have defined a function to change these values and attach object data to graphic objects in the following form.

*(feltolt layer\_name table field recnum value)*

With the help of this function the desired value can be stored in a (new) record's field of a specific table attaching to the objects situated on a given layer. If an objects has no or not enough record of the table, then the function attaches a new record to the object before setting its field value.

The basic idea to the realising the part-topology in practice was provided by the following observation. If the objects of a topology are available both in the project and the source drawings then it can be selected where to load the topology information to the memory from. The parameters of the VLISP function used to load a topology from an application are able to determine the origin of the data also.

In case of loading the topology from the source drawing the set of object cannot be controlled. But, if we can achieve that only a subset of the objects defining the topology would be copied to the project, which topology information is correct and complete when auditing, then there is no more difficulty to overcome.

The structure of layers in the source drawing and the function

*(geometria\_betolt list\_of\_layers)*

defined myself make it possible to define a property query, which always copies the objects belonging to the part-topology determined by the user's settings to the project drawing. So the analysing tools can be applied with success in the project.

As a result of the analysis a new network topology called *ut* is created in the project drawing, while the information about the route is stored in an object data table called BR\_ut. The automatic highlight of the result could be requested in the profile file controlling the analysis, but I thought it was better to develop a new function for this purpose. The function

*(utvonal\_megjelenitese)*

uses different colours and line types to represent the suggested route on the layer Utvonal. A legend appears on the drawing area, which helps to identify the types of public transport taken.

To improve the efficiency of application I had to ensure somehow the portability of the derived information. That is why I appended additional two submenus to the menu system of the application, which starts printing a report or a part of the map. The brief report can be appears on the screen in a dialogue box with a button staring printed the map.

### **2.3. Further functions of the application; custom solutions**

The target points (usually buildings) to visit during the route are to be selected from a larger set. This must be promoted as much as possible by a user friendly application. To fulfil this requirement a brief description appears automatically about the item highlighted in any list box of the dialogue box. What is more after pushing a button a photo is displayed about the facility to help the decision of the user.

The fast access of the topologic identification number of nodes selected by the user in the dialogue box as a target to visit is vital to execute the proper analysing tools. I created for this a text style with a specified structure

(*latniv\_hu.txt*), where each line stores the following data separated by a unique (!) space about a facility:

- Five-letter-long identity of the building
- Four-digit-long identification number of node,
- The name of the building (in language indicated in the name of the file)

The data read from the text file is formed into a list with specific structure. The items of it are created from data stored in a line of the text file, following the rules discussed in the dissertation. Each item contains a numeric value as well, describing which tiles of dialogue box belongs it to. This program variable and two further functions developed by myself make possible to watch the move of items between the two list boxes. One advantage of this structure is, that it is enough to ensure the alphabetical sorting of building's names in the text file in order to realise the right sort in the list boxes.

One of the endpoints of the links leading to buildings is always situated in the polygon representing the buildings. This makes possible the usage of the function

(*latniv\_keszit*)

which applies the point in polygon test, and which makes automatic the creation of text file mentioned above, or its update procedure in case of redefining the topology.

I completed the application with a user-friendly function as well that helps to determine the exact location of a street. After selecting the name of the street from a list, the corresponding objects become highlighted, and the view of the drawing changed depending on the objects' visibility.

The application developed in language called Visual LISP and can be found on the CD enclosed to the dissertations communicates with the users in two languages (Hungarian and English) at the moment. To change the language I defined a separate toolbar, which helps to modify the language of the application. The data depending on the current language (names, short guides) must be stored in all necessary languages, but it is sufficient to ensure the access only to the data in the current language.

The most suitable solution of the problem of this type is to store the data in separate text files with specific structure. The data in the current language can be read from the appropriate file into a program variable of list type by the application. Following the instructions detailed in the dissertation the scope of the available languages can be widened or restricted arbitrarily without any change in the source code of the application.

During developing the model and the application I made extra efforts that the graphical database could be exchanged at a later date, by not only the developer. As a result of this purpose sufficient information and instructions are available in the dissertation to build the physical model again. So any new source drawing about a settlement can be attached to the project.

### **3. The *point in polygon* test**

The discussing of the algorithm can be found in many books about basic knowledge of geographic information system, but unfortunately in a not appropriate form. The basic idea of the following algorithm and the ones mentioned above is identical, but it contains custom solution as well widening the scope of its usage.

The basic problem is the following:

Let us consider a polygon and a point. We must decide whether the given point is internal to the polygon or not.

The basic idea helping to answer the question:

Let us count, that how many times the vertical half-line starting from the given point and running to the positive direction intersects the sides of the polygon. If this number is even, then the point is external, if it is odd the point is proved to be internal.

The basic idea gives no instructions about the case when a vertex of the polygon is situated exactly on the vertical half-line. In this situation my conception was that the position of polygon sides belonging to the vertex compared to the half-lines starting from the point decides whether to increase the number of intersections or not. This way the counter must increased only if the mentioned sides are situated on the different side of the half-line. The vertical sides of the polygon –parallel to the half-line– can be omitted during the computation.

This specified algorithm works properly in the most general cases as well, like in case of concave polygons, non-simple polygons (it has got pair of non-consecutive edges sharing a point) or polygons with isolated islands. These special polygons can be found in practice as well if we think about the buffers, or the maps of countries containing a mini-state (like Italy contains San Marino) or ground-plan of buildings with open inner courtyard (The Reformed College in Debrecen).

The version of my algorithm implemented in Visual LISP programming language and used by the application can be found in the following form

(pont\_poligonban *p polygon*)

in the file called VInf.lsp, where the coordinates of point object and the entity name of the polygon must be provided as parameters of the function. The returning value is *nil* if the point was external otherwise it returns T.

## 4. New AutoCAD commands

During using the software and developing the application I missed some services which I replaced with Visual LISP functions developed by myself and its AutoCAD command forms where it was use of it. Autodesk Map is developed as the GIS extension of AutoCAD, that means it contains the entire functionality of AutoCAD.

The selection set is an arbitrary set of objects depending on the selection type, which has been provided an internal identification number. If it has been created by commands or Visual LISP functions, then the view of the drawing cannot be altered to make visible the objects of the set. With the help of my new Visual LISP function

(zoom\_kiv\_hz *sel\_set\_id*)

the zoom to the extension of objects in the selection set can be realised from applications.

The view of project drawing determines the set of objects of layers with property 'on' which are visible on the drawing area. This can be changed either manually or by altering the extension of source drawings. In this last case all the layers of the source drawing form automatically its extension, so we cannot zoom to the extension of a single layer. To replace this missing service I defined a VLISP function

(zoom\_folia *layer\_list*)

which set the view according to the extension of layers given as the parameter of the function. The AutoCAD command version of the function is defined in

zoom\_f

form, where the list of layers is created by selecting the items of a list box situated in the appearing Selecting layers dialogue box.

Removing of all the objects in arbitrary layers can be achieved by a single step using my Visual LISP function

(urit *folia\_lista*)

which command version is available in the following form.

urit\_f

The returning value is T, if the removing of the desired objects was successful, otherwise it returns *nil*. The list of layers for the command is created by the help of the dialogue box introduced at the former command.

The source codes of all functions and commands discussed in this chapter are available in the file *rutin.lsp* situated on the CD enclosed to the dissertation. The installation procedure of the application these commands are loaded, so they can be used immediately. If we need only the commands without the application, then it is sufficient to load the file called *rutin.lsp*.

Finally I stress the importance that the realisation of the analyses the part-topology created from an inhomogeneous network topology increases the power of analysing tools, its application and it makes possible to execute more complicated investigations. The new AutoCAD commands promote this procedure well and can be used even independent of the main results of the dissertation not only in GIS environment.

## Irodalomjegyzék (References)

- [1] AutoCAD Developer Documentation, Autodesk, Inc. 2002.
- [2] AutoCAD Map 2000 Felhasználói Kézikönyv, Autodesk, Inc. 2000.
- [3] Autodesk Map Getting Started, Autodesk, Inc. 2002.
- [4] Autodesk Map Visual LISP/ADSRX API, Autodesk, Inc. 2002.
- [5] BALLA-TÓTH-TRIPSÁNSZKY-NÉMETH: Térinformatikai szakági programozás, Műegyetemi Kiadó, 1995.
- [6] BARTELME, N.: Geoinformatik, Springer, 1995.
- [7] DAVID MARTIN: Geographic Information Systems, Routledge, 1996.
- [8] DETREKŐI-SZABÓ: Térinformatika, Nemzeti Tankönyvkiadó, 2003.
- [9] KERTÉSZ ÁDÁM: A térinformatika és alkalmazásai, Holnap Kiadó, 1997.
- [10] TIKÁSZ-KRAUTER-UGRIN-CSORNAI: A digitális térkép geometriai alapjai, Műegyetemi Kiadó 1995.
- [11] KOLLÁNYI-PRAJCZER: Térinformatika a gyakorlatban, GeoGroup Bt. 1995.
- [12] MACZKÓ-NAGY: LISP, AutoLISP programozás AutoCAD-ben IBM PC-n, LSI 1989.
- [13] NCGIA Core Curriculum Bevezetés a térinformatikába, (Szerk. Márkus B. Goodchild, M.F.–Kemp, K.K. nyomán), Székesfehérvár, EFE FFFK 1994.
- [14] NCGIA Core Curriculum Térinformatikai alapismeretek, (Szerk. Márkus B. Goodchild, M.F.–Kemp, K.K. nyomán) Székesfehérvár, EFE FFFK 1994.
- [15] NCGIA Core Curriculum Térinformatikai alkalmazások, (Szerk. Márkus B. Goodchild, M.F.–Kemp, K.K. nyomán) Székesfehérvár, EFE FFFK 1994.
- [16] NCGIA Core Curriculum Térinformatika Magyarországon, (Szerk. Márkus B.) Székesfehérvár, EFE FFFK 1994.
- [17] NCGIA Core Curriculum Térinformatikai Értelmező Szótár, (Szerk. Márkus B.) Székesfehérvár, EFE FFFK 1996.

- [18] STEIN, DAVID M.: The VisualLISP Developers Bible, 2003. Elérhető (Available): [www.dsxcad.com/book](http://www.dsxcad.com/book)
- [19] ZICHAR M.: Operations of Topology in GIS, *Proceedings of Seminar "Geometry and Graphics in Teaching Contemporary Engineer"*, Wisla 2000.
- [20] ZICHAR M.: Overlay Operations in GIS, *Proceedings of the 4<sup>th</sup> International Conference on Applied Informatics, Eger-Noszvaj*, 1999. Pages 217-222
- [21] ZICHAR M.: GIS and Computer Graphics, *Geometria i grafika inzynierska z.2 (Journal of Silesian Institute of Technology)*, Gliwice 1998. Pages 35-43

## **Publikációk listája (List of Publications)**

1. M. Zichar: GIS in Practice, *Scientific Journal of Silesian Technical University* (megjelenés alatt)
2. M. Zichar: Operations of Topology in GIS, *Proceedings of Seminar "Geometry and Graphics in Teaching Contemporary Engineer"*, Wisla 2000.
3. M. Zichar: Overlay Operations in GIS, *Proceedings of the 4<sup>th</sup> International Conference on Applied Informatics, Eger-Noszvaj, 1999*. Pages 217-222
4. M. Zichar: GIS and Computer Graphics, *Geometria i grafika inzynierska z.2 (Journal of Silesian Institute of Technology)*, Gliwice 1998. Pages 35-43
5. M. Zichar: A tutorial program, *Proceedings of the 2<sup>nd</sup> International Conference on Applied Informatics, Eger, 1995*. Pages 165-169

## **Egyetemi oktatási segédanyag (Tutorials)**

1. Zichar M: Oktatási segédlet az AutoCAD Map 2000 programhoz, 2000. pp.66.