

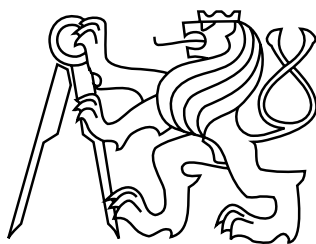
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ

DIPLOMOVÁ PRÁCE

PRAHA 2010

Bc. Zdeněk ZIMMERMANN

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ
OBOR GEODÉZIE A KARTOGRAFIE



DIPLOMOVÁ PRÁCE
TVORBA UŽIVATELSKÉHO ROZHRANÍ FUNKCÍ GIS
V SYSTÉMU MISYS

Vedoucí práce: Ing. Petr Soukup, Ph.D.
Katedra mapování a kartografie

leden 2010

Bc. Zdeněk ZIMMERMANN

ZDE VLOŽIT LIST ZADÁNÍ

Z důvodu správného číslování stránek

ABSTRAKT

Uživatelské rozhraní je to první na systému, co člověka zaujme. Navíc může určovat, zda bude systém používán a oblíben. Diplomová práce se zabývá návrhem rozhraní a jeho realizací u nových GIS funkcí systému MISYS. V první části práce jsou stručně zmiňovány funkce GIS a jejich rozdělení, dále pak uživatelské rozhraní systému MISYS. V dalších kapitolách jsou popisovány nově vzniklé funkce a návrh na rozšíření systému o funkce nové. Výsledkem diplomové práce je tedy nová funkce, která je součástí systému a je využívána v praxi. Ostatní funkce budou zahrnuty do systému až po jejich otestování.

KLÍČOVÁ SLOVA

systém MISYS, GIS, uživatelské rozhraní, dialog

ABSTRACT

Creation of user interface of GIS functions in MISYS system

User interface is the first thing that holds everybody's interest about the system. Moreover, it can determine whether the system will be used and popular. The diploma paper deals with design of an interface and with its realisation in new GIS functions of MISYS system. In the first part of the paper, functions GIS and their division are briefly mentioned and further user interface of MISYS system. In the following chapters, newly designed functions are described and a proposal for expansion of the system by new functions. The result of the diploma paper is a new function which is a part of the system and is used in the profession. The remaining functions will be included into the system after their testing.

KEYWORDS

MISYS system, GIS, user interface, dialog

PROHLÁŠENÍ

Prohlašuji, že diplomovou práci na téma „Tvorba uživatelského rozhraní funkcí GIS v systému MISYS“ jsem vypracoval samostatně. Použitou literaturu a podkladové materiály uvádím v seznamu zdrojů.

V Praze dne

.....

(podpis autora)

PODĚKOVÁNÍ

Chtěl bych poděkovat vedoucímu práce Ing. Petru Soukupovi, Ph.D. za připomínky a pomoc při zpracování této práce. Dále bych chtěl poděkovat panu řediteli společnosti GEPRO spol. s r.o. Ing. Zdenku Hoffmannovi, že mi umožnil pracovat na diplomové práci v této společnosti, stejně tak i dalším pracovníkům společnosti za jejich odborné připomínky. V neposlední řadě patří dík celé rodině za podporu ve studiu a především mé manželce, za trpělivost při psaní této práce.

Obsah

Úvod	8
1 Stručně o GIS	9
1.1 Rozdělení dat	9
1.2 Přehled funkcí GIS	11
2 Uživatelské rozhraní v systému MISYS	13
2.1 Menu	14
2.2 Panely	15
2.3 Grafické okno	16
2.4 Protokol	18
2.5 Místní nabídka	19
2.6 Transparentní funkce	19
2.7 Dialog	20
2.8 Vstupní tabulka	24
2.9 Horké klávesy	24
2.10 Stavový řádek	25
3 Návrh rozhraní Obalové zóny	26
3.1 Výstupní výkres a vrstva	27
3.2 Jednoduchá zóna	28
3.3 Multizóna	30
3.4 Generovat pro	32
3.5 Metoda zadání	32
3.6 Schéma výstupu	34
3.7 Spojit zóny a přenést atributy	35
3.8 Poslední řádek	36
3.9 Výsledky obalových zón	37
4 Atributová tabulka	39
4.1 Vznik tabulky a její plnění	40

4.2	Funkčnost ikon a tlačítek	43
5	Funkce překrytí (overlay)	51
5.1	Seznam funkcí	51
5.2	Funkce Rozpuštění (Dissolve)	52
5.3	Možnost rozšíření o další funkce GIS	57
5.3.1	Sjednocení (Union)	57
5.3.2	Ostatní funkce	58
	Závěr	60
	Použité zdroje	62
	Seznam symbolů, veličin a zkratk	63
	Seznam příloh	64
A	Příloha - ukázka zdrojového kódu	65

Úvod

Diplomová práce s názvem *Tvorba uživatelského rozhraní funkcí GIS v systému MISYS* byla vyhotovena v rámci magisterského studia oboru Geodézie a kartografie na ČVUT v Praze.

V dnešní době se stále častěji pracuje s daty, která mají určitý vztah k poloze na zemském povrchu. Díky tomu se oblast geografického informačního systému stále více rozrůstá, a tudíž je stále častěji využívána v nejrůznějších oborech. Tím rostou i požadavky na funkce v aplikacích. Uplatnění geodézie v procesu geografického informačního systému (dále jen GIS) vidím především ve fázi pořizování dat, s kterými pak bude dále pracováno. Nicméně i analytické nástroje mohou být v mnoha případech využívány při práci geodeta.

Cílem této práce je tedy navrhnout uživatelské rozhraní některých funkcí, které by byly využívány v GIS. Pro doplnění o tyto funkce byl vybrán systém MISYS, jednak proto, že žádnou z tvořených funkcí neobsahuje, nicméně je využíván při pořizování dat a tyto funkce jistě najdou uplatnění. Dále pak proto, že jeho součástí je vlastní programovací jazyk KOKEŠ Basic, ve kterém byly vyhotoveny už některé funkce zmiňované v mé bakalářské práci.

Diplomová práce je rozdělena do pěti kapitol. V první kapitole je stručně popsáno rozdělení dat a funkcí GIS. Druhá kapitola se zabývá uživatelským rozhráním systému MISYS. V dalších třech kapitolách jsou uvedeny nově navržené funkce společně s jejich popisem. Práce je doplněna obrázky, které nejlépe vystihují popisovanou problematiku. U funkcí Obalové zóny, Rozpuštění či Sjednocení bylo v rámci práce řešeno pouze uživatelské rozhraní a vše, co souvisí se zadáním dat uživatelem. Konečná data byla následně předložena výkonné funkci (v případě Rozpuštění a Sjednocení teprve budou předložena), která není předmětem této práce. Atributová tabulka je řešena v celém svém funkčním rozsahu, protože zde je uživatelské rozhraní s výkonnými funkcemi silně propojeno.

Kapitoly kromě vybraných ukázek neobsahují zdrojové kódy řešených funkcí. Pro ukázkou je uveden zdrojový kód funkce Rozpuštění, a to jak v tištěné podobě (viz příloha), tak digitálně na přiloženém nosiči.

1 Stručně o GIS

Oblast GISu je široká a různorodá. Cílem této práce však není rozebrat GIS do všech jeho detailů, protože této problematice bylo věnováno velké množství publikací, např. *Geografické informační systémy 10¹* [1] či *Understanding GIS* [2] nebo *Úvod do geografických informačních systémů* [3] a další. Přesto je zde několik řádků věnováno typům dat a gisovským funkcím, jejichž znalost je důležitá pro lepší orientaci v následujících kapitolách.

1.1 Rozdělení dat

Data se v GIS aplikacích vztahují k tzv. *objektu*, což je dále nedělitelný prvek a je základem objektově orientovaného programování [4]. Každý objekt musí mít svůj *identifikátor*, což je jeho jedinečné číslo. Data lze rozdělit na data *geometrická*, vztažená k určité poloze a data *negeometrická*, která doplňují popis objektu. V obou případech se však jedná o data prostorová.

Geometrická data

Obsahují informace o poloze určitého krajinného prvku a jsou vyjadřována velikostí souřadnic v předem daném systému. Údaj o poloze objektu v obrazové podobě je znázorněn *geometrickými útvary* (obr. 1.1) [1]:

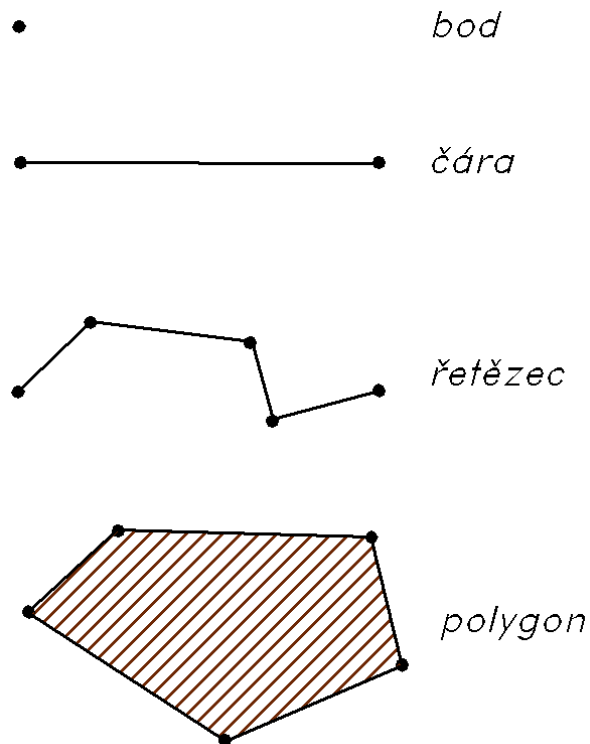
- a) body,
- b) čáry,
- c) plochy.

Skupina čar stejného typu na sebe vzájemně navazujících je nazývána *řetězcem* (obr. 1.1). Plochy jsou v GIS terminologii označovány jako *polygony*, které jsou tvořeny nejméně třemi body, přičemž dva musí být totožné.

¹Prvně zmíněné pojmy či speciální výrazy budou vždy v textu zvýrazněny kurzívou či jiným druhem písma.

Negeometrická data

Jedná se o atributy vztažené k objektu podle příslušného identifikátoru. V některých případech může být třetí souřadnice považována za atribut (např. u vrstevnic). Negeometrickými daty jsou však i údaje popisné, které jsou naznačeny vizuálně, ať barvou, výplní či textem.



Obr. 1.1: Geometrické útvary prostorových dat.

Prostorová data potřebujeme nějak převést do digitální podoby. Využíváme k tomu dva přístupy odlišné způsobem kódování:

- **vektorový model** - výchozím geometrickým útvarem je bod,
- **rastrový model** - základní kódovací jednotkou je buňka v síti.

Oba modely lze dále dělit podle přístupu uspořádání dat na další typy, nicméně tím se dále nebudu zabývat. Obecně lze ke každému modelu říci, že vektorová data jsou vhodnější pro přenášení informací pomocí čarových elementů, rastrová jsou naopak vhodná pro prezentaci ploch. GIS by měl proto využívat obou modelů.

1.2 Přehled funkcí GIS

Součástí GIS je i funkční vybavení, které se nazývá *gisovskými aplikacemi*. Jedná se o sadu funkcí daného softwaru pro zpracování vybraných dat. Funkce můžeme podle typu operace zařadit do čtyř následujících skupin [1]:

1. funkce pořizování dat,
2. údržbové a manipulační funkce,
3. funkce na analýzu dat,
4. výstupové funkce.

V aplikaci nemusí být zastoupeny všechny druhy funkcí, naopak aplikace může mít více funkcí stejného typu. Obecně lze říci, že GIS aplikace od firmy ESRI se stala v tomto oboru standardem, a proto mnoho produktů a jejich řešení je s ní porovnáváno. Dále budou zmiňovány pouze funkce pro analýzu dat, protože právě tyto funkce jsou dále obsahem diplomové práce.

Funkce na analýzu dat

Pro analýzu dat jsou používány takové funkce, které uživateli systému umožní získat potřebnou informaci o části krajiny. Komunikace s analytickými funkcemi je zajištěna pomocí tzv. *dotazů*. Analytické funkce pak můžeme podle prováděné operace rozdělit na funkce *výběrové*, *výpočtové* a *modelovací*.

Výběrové funkce slouží k dotazování jak na geometrická, tak popisná data. Proto se tento typ funkcí nazývá též *funkcemi dotazovacími*. Základní funkcí v této kategorii je dotazovací funkce na atributy. Funkce nepracuje složitě s polohovými daty a topologickými vazbami, nicméně umožňuje efektivně vyhledávat v attributech, třídit je a podobně. Data jsou zobrazena většinou pomocí přehledné tabulky. Příkladem takové funkce je Atributová tabulka (viz kap. 4).

Výpočtové funkce se používají na matematické, logické, geometrické a statistické operace s atributy nebo polohovými daty. Do této skupiny patří jak jednoduché funkce pro součet, maximum aj., dále však i funkce složitější odpovídající na dotazy ohledně spojení či cesty mezi dvěma místy.

Modelovací funkce je skupina funkcí nejsložitějších. Výsledky těchto funkcí se také nazývají někdy jako *předpovědní*, protože dávají odpověď, co se stane či by se stalo na jiném místě nebo v jiném čase.

Použití výběrových funkcí je často pouze první částí analýzy, v další fázi jsou data zkoumána právě výpočtovými a modelovacími funkcemi, které lze rozdělit do čtyř kategorií:

1. klasifikační a měřicí funkce,
2. funkce překrytí²,
3. funkce v okolí,
4. spojovací funkce.

Klasifikační a měřicí funkce jsou takové funkce, které pracují s oběma typy dat (geometrickými i negeometrickými), nicméně mění jen atributy - nemění se poloha a nevznikají nové objekty. Účelem funkce překrytí je získat ze dvou souborů vzta-hujících se ke stejnému území nový datový soubor obsahující nové objekty. Funkce v okolí se používají pro zjištění vlastností určité oblasti kolem zadaného objektu. Spojovací funkce zjišťuje hodnotu určitého atributu nebo počítá velikost odvozeného parametru z polohových a popisných dat. Pod tento typ funkcí patří i funkce obalové zóny (buffer), která bude více řešena v kapitole 3.

V dnešní době není používání GIS výhradně výsada konkrétního oboru. Díky různorodosti použití, rychlého rozvoje a nízké ceně počítačové techniky může být využí-ván v podstatě kdekoli. Příkladem může být veřejná správa, oblast přírodních zdrojů a inženýrských sítí, doprava, obrana státu, kartografie a mnoho dalších oborů [5].

²Další popis této kategorie funkcí je obsahem kapitoly 5.1.

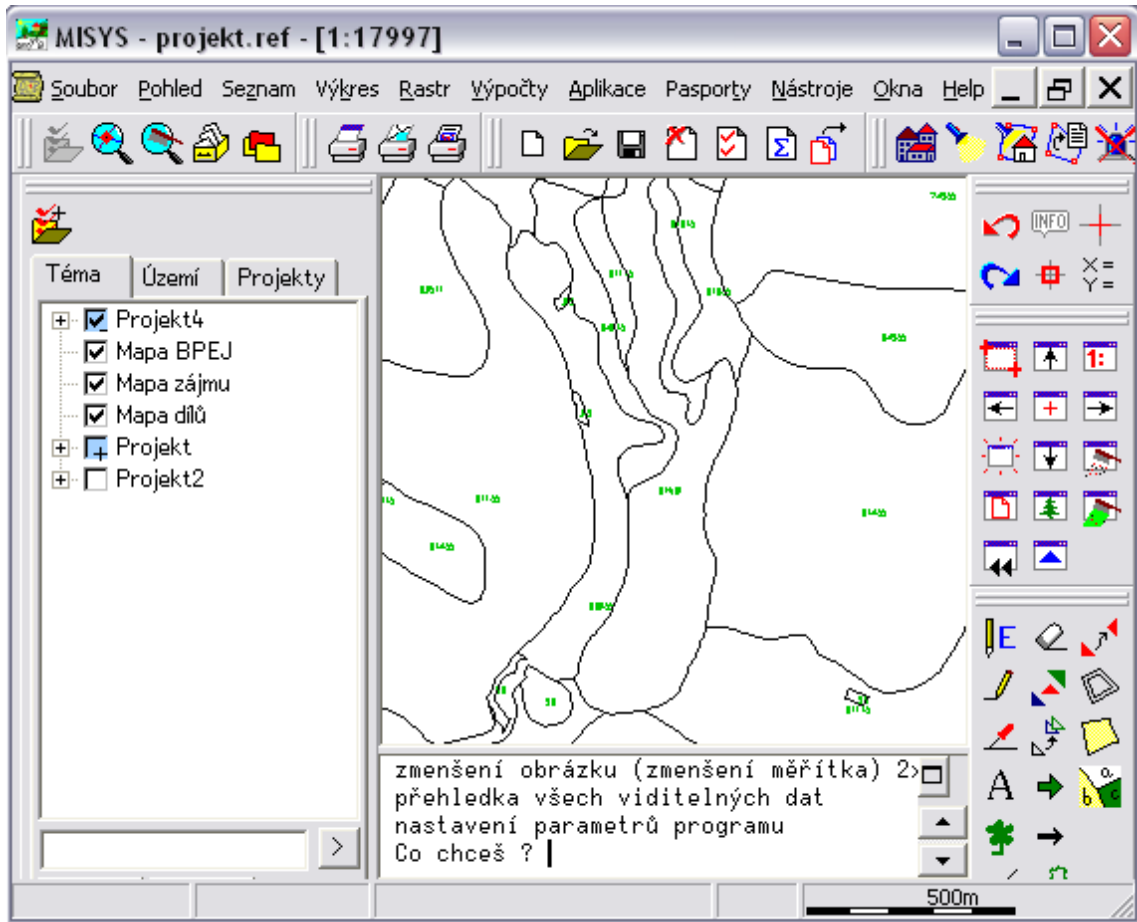
2 Uživatelské rozhraní v systému MISYS

Vhodné uživatelské rozhraní je důležité pro každý systém. Ten je totiž složen z výkonných funkcí, které provádějí nejrůznější operace. Je proto třeba zajistit takové rozhraní, aby funkce mohl uživatel jednoduše a efektivně využívat. Důležité je si také uvědomit, že ovládání a vzhled rozhraní často rozhodují, zda je program používán a je oblíbený.

Na systému MISYS je zajímavé, že není stavěn jako jiný klasický GIS software. Společně s produktem PROLAND¹ sloužícím pro zpracování pozemkových úprav, vychází jeho prostředí a funkčnost z produktu KOKEŠ, který je vhodným nástrojem pro geodetické práce a pro tvorbu a údržbu mapových děl. Na jednu stranu to může být omezující faktor, na druhou však značná výhoda. Základy systému MISYS jsou tedy převzaty, dále je však software tvořen tak, aby splňoval požadavky GIS aplikace. Navíc může být doplněn o aplikaci KATASTR sloužící pro práci s databází SPI KN, dále o aplikaci PASPORTY, která umožňuje uživatelům připojit do své instalace řadu aplikací z různých oborů [6]. Vznik nových funkcí systému a úpravy starých jsou často prováděny podle potřeb uživatelů, čímž je zaručeno propojení s praxí, a tudíž i efektivnost systému.

Systém se skládá z klasických prvků, které jsou pro tento typ programů typické. Jedná se jednak o *menu* s nabídkami dostupných funkcí, *grafické okno* pro práci s různými typy dat, *protokol* pro komunikaci s uživatelem a pro výpis potřebných dat, *panely*, *místní nabídky*, *transparentní funkce* na pozadí systému, *horké klávesy*, a v neposlední řadě o *dialogy*. Zvláštním prvkem může být tzv. *vestupní tabulka*. Část prvků je vidět na obrázku 2.1. V následujících podkapitolách budou probírány právě tyto prvky.

¹MISYS, PROLAND a KOKEŠ jsou všechno produkty společnosti GEPRO spol. s r.o.



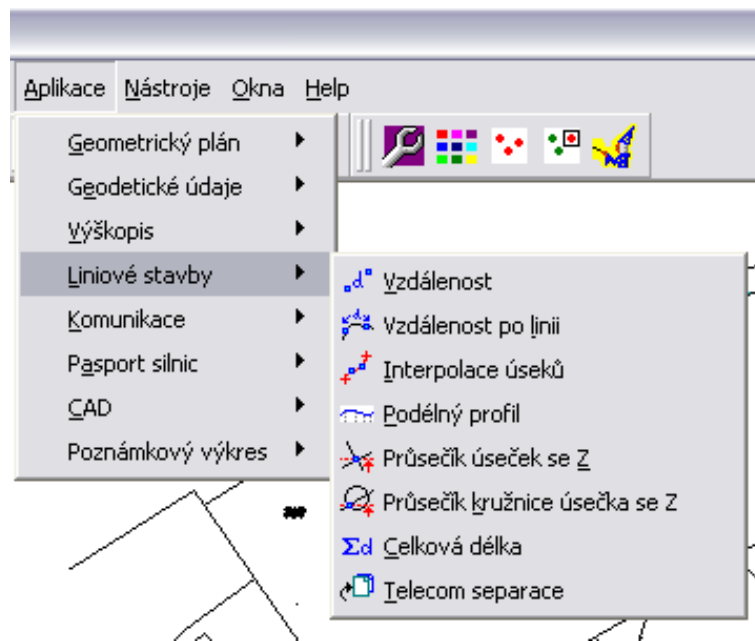
Obr. 2.1: Systém MISYS

2.1 Menu

Vpodstatě každý obsáhlejší systém má menu, které obsahuje většinu dostupných funkcí systému. Menu je speciálním případem panelu (viz níže). Oproti klasickému panelu však nevolá konkrétní funkci, ale rozbaluje nabídky dostupných funkcí, které se tématicky váží na jméno položky menu. Nabídky mohou teoreticky obsahovat libovolný počet položek, které mohou mít opět svoje podnabídky². Aktuálně nedostupné funkce jsou obvykle v nabídkách znepřístupněny. Ovládání menu je možné buď klasicky stiskem levého tlačítka myši nebo lze k položkám menu přistupovat pomocí klávesnice. Po stisku klávesy *Alt* jsou příslušná písmena položek podtržena,

²Obvyklý počet vnoření je tak do třetího stupně.

a pak tedy lze k položkám přistoupit přes stisk klávesy *Alt* a označeného písmene. Pohyb v nabídce je pak uskutečňován pomocí šipek. Uspořádání položek menu a jeho nabídek je velice důležité pro intuitivnost a efektivnost. Často se však k funkcím nepřistupuje přes menu, ale existují nejrůznější příkazy či zkratky, které volání požadované funkce urychlí, což usnadňuje práci. Příklad menu je vidět na obrázku 2.2.



Obr. 2.2: Menu

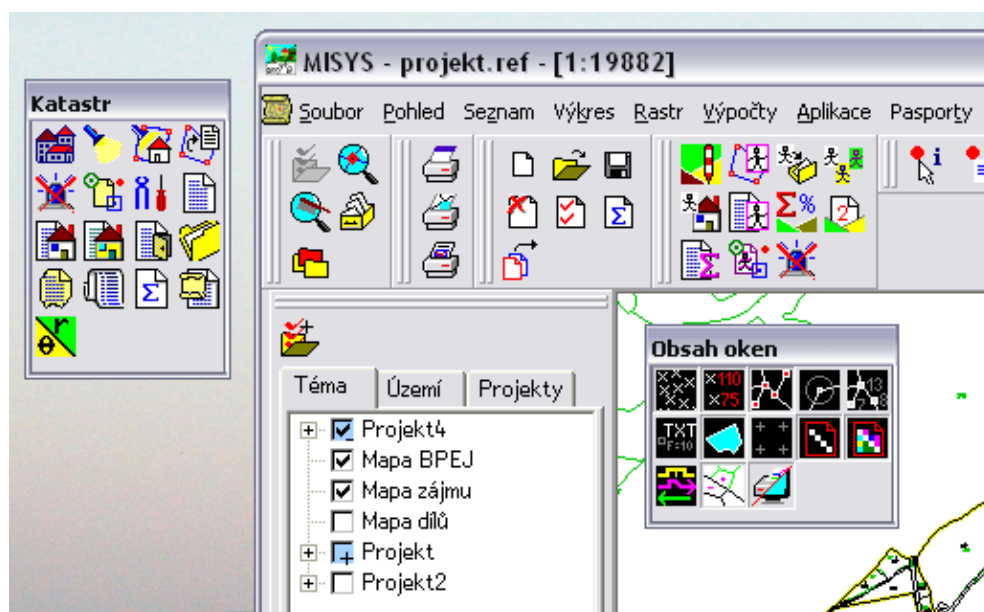
2.2 Panely

Panel je prvek obsahující libovolný počet položek. Ty jsou znázorněny zpravidla obrázkem (ikonou) a po stisku volají požadovanou funkci. V panelu může být i prvek dialogu, nejčastěji se jedná o nabídku (combobox). Pořadí položek se obvykle nemění³, nicméně je možné upravovat tvar panelu, a také jeho umístění - panel buď může být v pravé či levé části systému, dále pak nahoře pod panelem. Pokud bychom chtěli v rámci systému panel umístit na jiné místo, musíme panelu nastavit, aby se choval jako *volné okno*. Při volbě panelu jako *samostatné okno* lze panel umístit na

³Pro změnu pořadí položek lze využít volbu *Upravit prostředí*, která je ale určena spíše zkušenějším uživatelům.

jakékoli místo obrazovky, tedy i mimo systém MISYS. Panely lze během chodu systému libovolně zapínat a vypínat. Systém umožňuje vytvořit i panely nové, a to pomocí volby **Upravit prostředí**⁴.

Velkou předností panelů je jejich variabilita. Panely při dalším spuštění systému zůstávají ve stavu a místě, ve kterém byly před opuštěním systému. Používání panelů usnadňuje práci, není totiž potřeba funkce složitě hledat v menu. Některé panely a jejich rozložení znázorňuje obrázek 2.3.



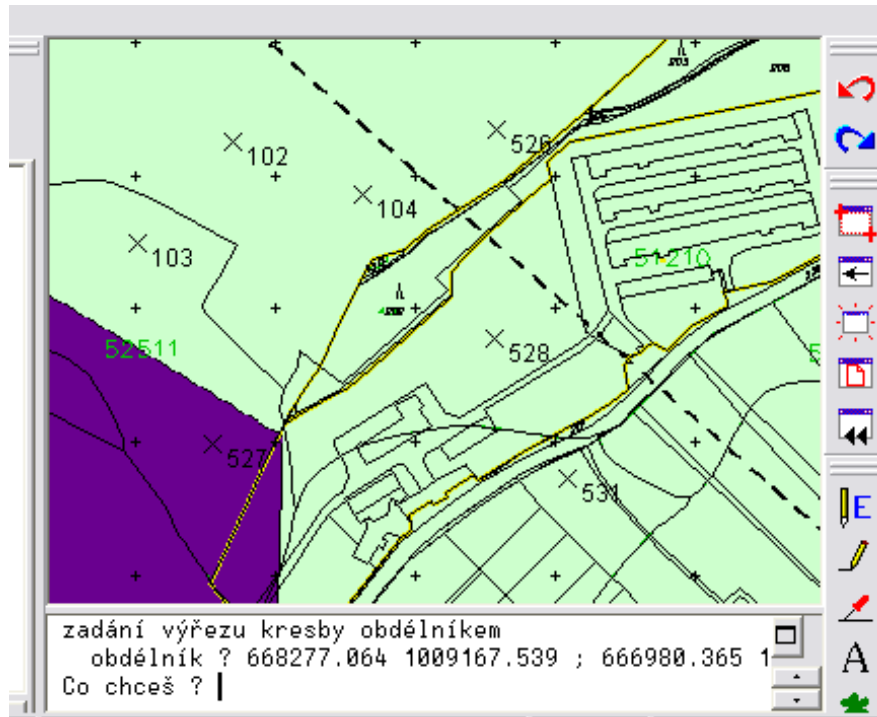
Obr. 2.3: Panely v systém MISYS

2.3 Grafické okno

Jelikož v oblasti GIS pracujeme hojně s grafickými daty, musíme mít možnost je někde prezentovat. Grafické okno je tedy oblast systému, ve kterém se tato data zobrazují. Data jsou uspořádána do souborů různých formátů. Mezi podporované formáty patří jednak výkresy, seznamy souřadnic či jiné formáty od společnosti GEPRO spol. s r.o., dále pak nejrůznější rastry a formáty jiných software pracujících s geografickými daty (MicroStation, ESRI, Autodesk).

⁴Volba **Panely** pro vypnutí a zapnutí panelu, dále volba **Upravit prostředí** jsou dostupné po stisku pravého tlačítka myši nad oblastí systému, kde se panely umísťují.

Důležité je podotknout, že otevřený soubor se nerovná otevřenému oknu, tzn. otevřené soubory nevyžadují grafické okno a naopak. Jelikož však velká část funkcí pracuje právě s daty navázanými na grafiku, je často existence okna nutná, navíc pro přehlednost práce velice důležitá. Pro ukázkou uvádím grafické okno s načteným výkresem (obr. 2.4).



Obr. 2.4: Grafické okno

Každé okno má své měřítko, které je zobrazováno na horní liště systému. Jednotlivým souborům lze nastavit jejich zobrazení v okně, a dále jakou budou mít vlastnost vůči oknu - jejich viditelnost, identifikaci a editaci (*Soubor/Vlastnosti*). Různé vlastnosti chování se dají nastavit i samotnému grafickému oknu, příkladem může být zobrazování informací v okně (*Pohled/Obsah oken*). Zbytek funkcí záložky *Pohled* jsou nástroje pro samotnou práci s oknem, ukázkou může být zmenšení a zvětšení výřezu okna, jeho překreslení či výřez. Všechny tyto funkce jsou funkcemi transparentními, a mnoho z nich je implicitně navázáno na horké klávesy⁵.

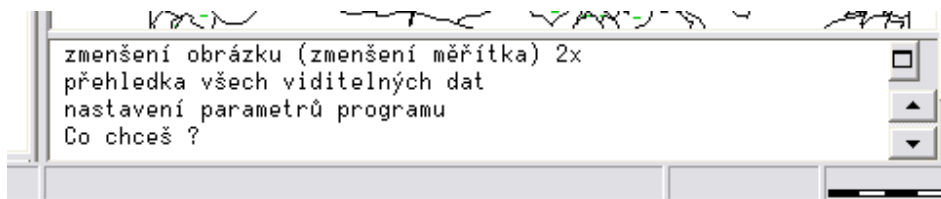
Pro práci s okny je také určena položka menu, a to s názvem *Okna*, která obsahuje nástroje pro založení nového okna (oken může být více) a zavření stávajícího, dále

⁵Význam transparentních funkcí a horkých kláves je popsán v následujících kapitolách.

nástroje pro uspořádání oken a v poslední části nabídky seznam oken s jejich měřítky, ve kterém jsou zrovna zobrazena. V okně nemusí být zobrazeny vždy soubory se vším svým obsahem, možné je vytvářet tzv. *selekcí*, která určuje, co bude v okně zobrazeno a co bude skryto. Selekcce se váže vždy k aktuálnímu oknu, tudíž lze v různých oknech provést různou selekcí. Vlastností okna je i to, že z něj lze identifikovat nejrůznější prvky, stejně tak je tam graficky vkládat.

2.4 Protokol

Protokol je jedním ze způsobů, jak systém komunikuje s uživatelem. Protokol má dva hlavní úkoly. Prvním úkolem je pouze informovat uživatele, tzn. není požadována interakce s uživatelem. Příkladem může být výpis informací při otevření dat v grafickém okně či chybová hlášení. Druhým úkolem je interaktivní komunikace při běhu některé z funkcí, kdy je požadováno zadat např. číslo bodu či jeho souřadnice. Tento způsob by mohl být zcela nahrazen dialogem, nicméně při požadavku na konkrétní neobsáhlou informaci urychluje komunikaci s uživatelem, a tudíž jeho práci.



Obr. 2.5: Protokol

Základním stavem protokolu je výpis textu *Co chceš?*, který nastává, pokud systém nepracuje a čeká na pokyn uživatele. V tomto stavu lze zavolat funkci její předdefinovanou zkratkou⁶. Velikost protokolu je možné měnit tažením horního okraje protokolu pomocí myši nebo stiskem klávesy *F2*, kdy je protokol zvětšen přes celé grafické okno. Oproti menu a stavovému řádku protokol nelze zcela skrýt - vždy je zobrazen alespoň jeden řádek protokolu. Po stisku pravého tlačítka myši nad protokolem je vyvolána místní nabídka s možnostmi práce s protokolem. Lze do protokolu

⁶Zkratky lze přiřazovat a měnit pomocí funkce *Upravit prostředí*, která již byla zmíněna.

zapsat vlastní text, tisknout či kopírovat jeho obsah, popřípadě ho plně editovat pomocí *Editoru protokolu*, kde je pak možné v protokolu vyhledávat či nahrazovat text, uložit protokol do textového souboru atp. Výpočty se též zapisují do protokolu mezi ostatní výpisy, nicméně pouze k editaci výpočtů je k dispozici *Editor záznamu*.

2.5 Místní nabídka

S určitým druhem nabídky jsme se už setkali u předchozích prvků. Způsob použití místní nabídky nebude popisován, jelikož je totožný s nabídkou použité u menu (kap. 2.1). Tento druh nabídky však není vázán na menu, ale na pravé tlačítko myši, což umožňuje nabídku vyvolat v libovolné části systému. Příkladem může být nabídka po stisku pravého tlačítka myši nad grafickým oknem (obr. 2.6) či protokolem.



Obr. 2.6: Místní nabídka

2.6 Transparentní funkce

Funkce v systému MISYS bych rozdělil do dvou kategorií. Do první kategorie spadají funkce, které v případě běhu některé předcházející funkce tuto funkci ukončí a spustí se právě nově volaná funkce. Druhým případem jsou funkce, které předcházející funkci pouze potlačí, vykonají svoji úlohu a zpřístupní původní funkci. Do této

kategorie patří právě tzv. transparentní funkce [7], které mohou proběhnout na pozadí bez dalšího dotazu (např. Přehledka všech dat, Zvětšení 2x a Zmenšení 2x dostupné z menu volba **Pohled**) nebo s konkrétním dotazem (např. Měřítko okna též z menu volba **Pohled**). Jak je možné z uvedených příkladů vyzorovat, v drtivé většině se jedná o funkce pro práci v grafickém okně. Zvykem je, že transparentní funkce se umísťují na klávesy *F*, a to jednak z důvodu četnosti jejich použití, dále pak proto, že při ukliknutí tyto funkce neukončí původní funkci, pokud je spuštěna.

2.7 Dialog

Dialog je velmi významným nástrojem pro komunikaci s uživatelem. Důležité je si uvědomit, že vzhled a správná funkčnost dialogu má vliv na oblíbenost funkce, která dialog používá. Dialogy v systému MISYS se dají rozdělit na dva druhy. Prvním je *dialog modální*, při jehož spuštění není možné přistupovat k jiným prvkům systému. Dialog musí být tedy nejdříve ukončen. Opačným případem je *dialog nemoďální*, při jehož běhu může uživatel přistupovat k jiným funkcím systému. Po proběhnutí nově zvolené funkce se buď systém vrátí nebo nevrátí k funkci původní, jak už bylo zmíněno při popisu transparentní funkce. Definice dialogu je uvedena v mé bakalářské práci [7]. Programátorsky je méně náročný dialog modální, není u něj potřeba ošetřovat případy, kdy uživatel zavřel nějaký soubor nebo změnil jeho vlastnost, popřípadě zavřel grafické okno atd. Pokud funkce nevyžaduje složitější zadání dat či nastavení, je výhodnější využít vstupní tabulku, vstup z protokolu či grafického okna. Složitější zadání se však bez dialogu neobejde.

Navrhnout vhodný dialog a vyřešit jeho chování je často náročnější, než vytvořit samotnou funkci, která bude požadovanou funkčnost provádět. Jelikož nelze podrobně popsat všechny prvky dialogu v rozsahu jedné kapitoly, uvádím v tabulkách 2.1 až 2.4 seznam prvků s jejich stručným popisem.

prvek	popis
areabox	rozbalovací seznam výplní ploch
cnrbox	seznam výkresů, seznamů souřadnic a rastrů formou tabulky
colorbox	rozbalovací seznam barev
combobox	rozbalovací seznam, který lze naplnit libovolným textovým obsahem
fontbox	rozbalovací seznam fontů (textových stylů)
linebox	rozbalovací seznam kreslicích klíčů
listbox	seznam, který lze naplnit libovolným textovým obsahem
pattbox	rozbalovací seznam vzorů čar
symbolbox	rozbalovací seznam symbolů
toolbox	panel nástrojů
widthbox	rozbalovací seznam šířek čar

Tab. 2.1: Nabídky (rozbalovací seznamy) dialogu

Obsah nabídek je plněn buď před voláním dialogu nebo při jeho vzniku. Uživatel nemá možnost obsah nabídek během trvání dialogu měnit.

prvek	popis
edit	editační pole pro zadání libovolného textu (může být víceřádkové)
editint	editační pole pro zadání celého čísla
editlist	editovatelný seznam - kombinace editu a comboboxu nebo listboxu
editreal	editační pole pro zadání reálného čísla
richedit	editační pole pro zpracování textu ve formátu RTF

Tab. 2.2: Editační pole dialogu

Jedná se ve všech případech o editovatelná pole. Zvláštností je editlist, který lze považovat za spojení editu a comboboxu - do pole editlistu lze psát vlastní text nebo vybrat jednu položku z nabídky. V editačních polích, kromě richeditu, nelze zpracovávat text (tzn. psát kurzívou či tučnějším písmem aj.).

prvek	popis
button	tlačítko
checkbox	zaškrtačací pole
radiobox	přepínací pole

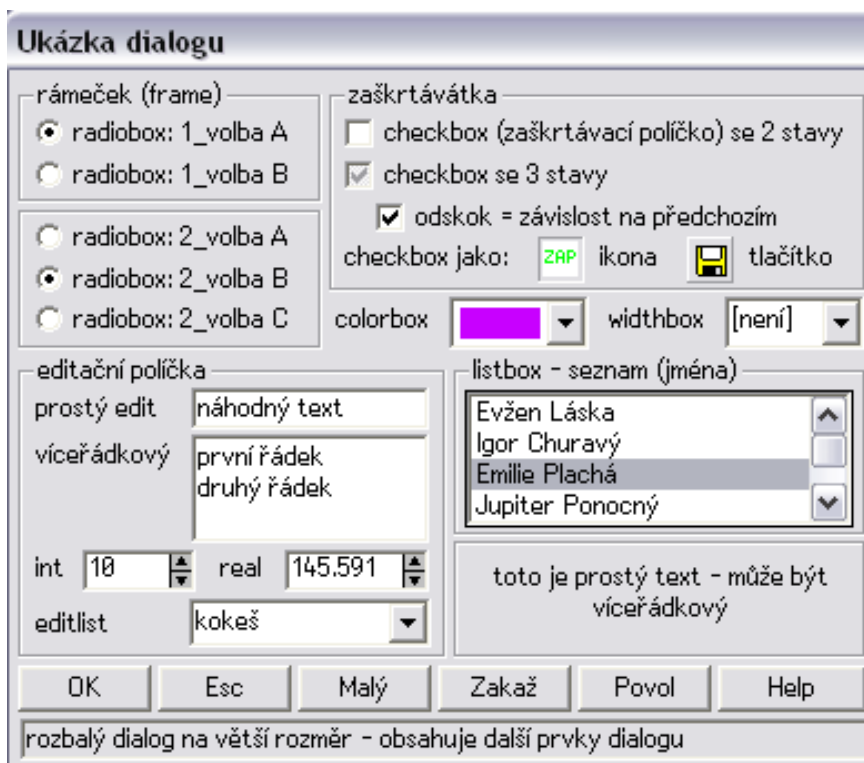
Tab. 2.3: Tlačítka dialogu

Radiobox je spojován do skupin, aby fungoval jako přepínač. Checkbox může být třístavový - vypnuto, zapnuto, znázornění závislosti (obr. 2.7).

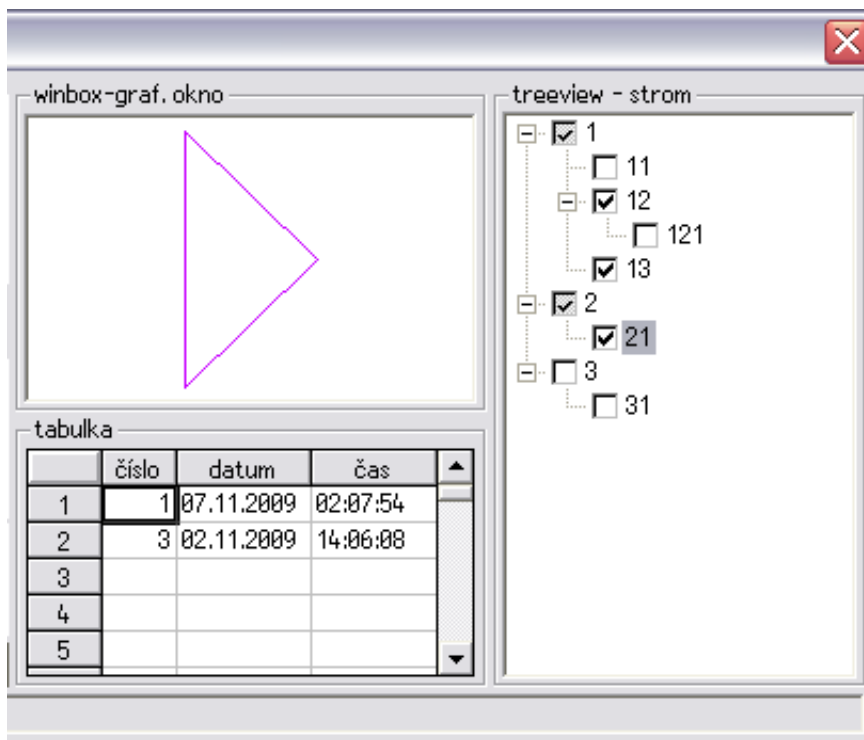
prvek	popis
control	obecný prvek (rodičovská třída pro ostatní prvky)
Form	informace o textovém editoru
frame	rámeček
icon	obrázek ve formě ikony
scrollbar	posuvník
tabledef	tabulka dialogu - sloupce mohou být různého datového typu
text	nadpisy - needitovatelný text
treeview	stromová struktura
userctl	obecný prvek dialogu - uživatelsky modifikovatelný
winbox	grafické okno

Tab. 2.4: Ostatní prvky dialogu

Zadané hodnoty v dialogu si lze pamatovat do dalšího spuštění funkce, a to pomocí *inicializačního souboru*, tj. souboru s příponou *ini*, umístěného ve stejném adresáři z kterého se spouští systém. Během chodu dialogu, tak i po jeho ukončení, může být volán další dialog. Pro ukázkou dialogu uvádím dialog, jehož vzhled je názorný z obrázku 2.7. Tento dialog lze rozšířit stiskem tlačítka *Velký* o další prvky (obr. 2.8). Ukázka obsahuje převážnou část prvků, které lze v dialogu použít. Nejedná se o používaný dialog systému, pouze o dialog ukázkový (zdrojový kód dialogu je součástí příloženého CD, takže je možné si funkci některých prvků vyzkoušet).



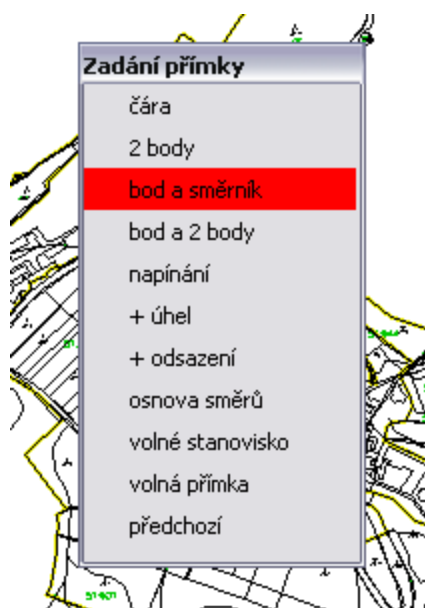
Obr. 2.7: Možnosti dialogu - část první



Obr. 2.8: Možnosti dialogu - část druhá

2.8 Vstupní tabulka

O vstupní tabulce lze říci, že je to velmi speciální druh dialogu. Obsahuje pouze řádky, které se chovají jako tlačítka a mají na sebe navázanou určitou funkčnost. Tlačítka se mohou chovat jako zapínač nějaké vlastnosti - v tomto případě může být označeno více řádků tabulky najednou (např. funkce Ortogonální metoda - řádek s nápisem *+ úhel*) nebo mohou vyvolat další dialog. Především se však chovají tak, že po jejich použití systém čeká na příslušný vstup (input). Podle této vlastnosti také název vstupní tabulka. Tabulka neobsahuje kromě řádků nadbytečné prvky. Požadovaný vstup lze provést buď výběrem v grafice nebo zadáním požadovaných údajů v protokolu, který tabulka používá výhradně pro komunikaci s uživatelem. Díky těmto vlastnostem je tabulka používána především u výpočetních funkcí v záložce menu *Výpočty*, dále pak při tisku, výpisu a identifikaci prvků. Ukázka vstupní tabulky je na obrázku 2.9.



Obr. 2.9: Vstupní tabulka

2.9 Horké klávesy

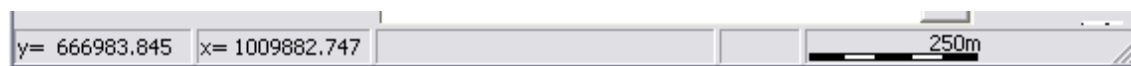
Horké klávesy jsou funkční klávesy *F2* až *F9*, *F11*, *F12*, a dále kombinace kláves *Ctrl/Alt/Shift+F1* až *F12*, *Ctrl+A* až *Z* (či libovolné písmeno), *Ctrl/Alt+Backspace*

a kombinace *Ctrl+Shift* nebo *Ctrl+Alt+výše uvedené*. Horké klávesy lze nastavit v *Úpravě prostředí*. Některé klávesy má vyhrazen operační systém, proto je nelze použít jako horké klávesy. Jedná se o klávesy:

- *F1* - vyvolání nápovědy,
- *F10* - skok do menu,
- *Alt+F4* - zavření aplikace.

2.10 Stavový řádek

Stavový řádek se nachází v dolní části systému. Jedná se o pruh, který zobrazuje různé informace, lze ho také jako v případě menu schovat. Uživatel nemá možnost měnit vzhled a rozložení buněk v řádku a nemůže do něj zapisovat jako u protokolu. Standardně jsou v řádku vypisovány souřadnice a měřítko (obr. 2.10), při nájedzu na funkci je zobrazen její krátký popis a zkratka, přes kterou lze funkci z protokolu spustit. Při běhu některé z funkcí zde mohou být zobrazovány nejrůznější údaje, například aktivní seznam souřadnic nebo tzv. *teploměr*, který vyznačuje průběh zpracování dat příslušnou funkcí.



Obr. 2.10: Stavový řádek

Systém MISYS obsahuje vlastní programovací jazyk *KOKEŠ Basic*, pomocí něhož lze vzhled a chování systému částečně přizpůsobovat vlastním představám, stejně tak lze jeho pomocí tvořit funkce nové. Následující funkce byly vytvořeny právě v tomto jazyku.

3 Návrh rozhraní Obalové zóny

Ačkoli systém MISYS obsahuje velké množství funkcí a nadstavby v podobě aplikací, poslední dobou je vyvíjen tlak na to, aby v systému vznikly nejrůznější analytické funkce. První tvořenou funkcí z této kategorie je právě *Obalová zóna*. Podle rozdělení funkcí v první kapitole bychom Obalovou zónu zařadili do skupiny spojovacích funkcí. Funkce slouží k vytvoření tzv. *obalové (nárazníkové, buffer) zóny*.

Funkce jako celek je rozdělena striktně na dvě části, a to na část řešící rozhraní mezi uživatelem a výkonnou funkcí a na samotnou výkonnou funkci. Výměna zadaných dat je uskutečněna pomocí *struktury*, která obsahuje jednotlivé *proměnné*.

Struktura předávaná výkonné funkci pro tvorbu obalové zóny:

```

record buffer_rec          'struktura
  container cnr            'cílový výkres
  string vrstva           'cílová vrstva
  string copy_atr         'seznam přenášených atributů oddělený středníky
  string atr_sirka        'atribut, ze kterého se má převzít šířka zóny (pouze
                          pro jednoduchou zónu; je-li vyplněn, má přednost před
                          šířkou zadanou hodnotou
  int flag                'viz BUFFER_*
  int pocet               'počet zón (1 pro jednoduchou zónu, 2 až N pro
                          více zón)
  real dynamic sirka[1]   'šířka zadaná hodnotou (délku pole a zda se tedy jedná
                          o více zón určuje proměnná pocet)
  int dynamic klic[1]     'hodnota kreslicího klíče (<=0 značí nekreslit obvod;
                          délku pole a zda se tedy jedná o více zón určuje
                          proměnná pocet)
  int dynamic plocha[1]   'hodnota výplně plochy (<0 značí nekreslit plochu;
                          délku pole a zda se tedy jedná o více zón určuje
                          proměnná pocet)
  point dynamic pnts[0]   'body při zadání dat oblastí
  int ipnts               'počet zadaných bodů při zadání dat oblastí
  identify dynamic ide[0] 'prvky vybrané při identifikaci dat pro tvorbu zóny
                          (objekt, bod, typ)
  int iide                'počet vybraných objektů
end record
  
```

Použité *vlajky*, jejichž hodnota je přenášena pomocí proměnné *flag*:

```
#define BUFFER_POINTS      1      'body
#define BUFFER_LINES      2      'linie
#define BUFFER_AREAS      4      'plochy
#define BUFFER_JOIN       8      'vytvořené zóny spojit
#define BUFFER_COPY_ATR   16     'přenést atributy
#define BUFFER_FENCE     32     'zadání oblastí
#define BUFFER_WHOLE     64     'spadá celý
#define BUFFER_IDENTIFY   128    'identifikace
#define BUFFER_OUT       256    'vně oblasti
#define BUFFER_IDENTICAL  512    'pouze shodné
```

Při úvaze nad řešením rozhraní pro tuto funkci se musí vycházet z možností, které systém umožňuje. Po srovnání s totožnou funkcí od společnosti ESRI je zřejmé, že přístup k vstupním a výstupním datům bude odlišný, a to proto, že uspořádání dat v obou systémech je jiné. Způsob řešení bude zřejmý z dalšího popisu obsahu dialogu.

Aby mohla být funkce spuštěna, je požadována existence grafického okna a výkresu, který má zapnutou vlastnost pro kreslení v grafickém okně. Zbývající dvě vlastnosti výkresu nejsou implicitně pro spuštění požadovány, ale pokud výkres nebude identifikovatelný, nelze provést zadání dat identifikací, při neidentifikovatelnosti nelze výsledek uložit do tohoto výkresu. Po spuštění funkce je nabízen její hlavní dialog, jehož části si popíšeme v jednotlivých podkapitolách.

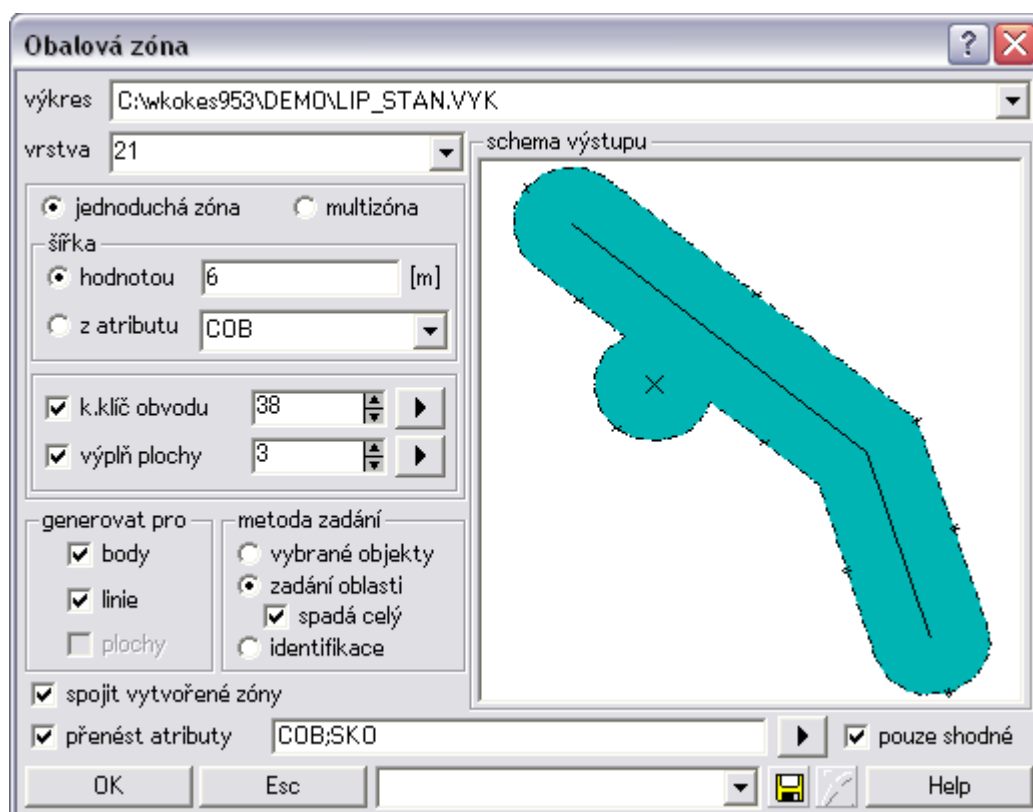
3.1 Výstupní výkres a vrstva

Výkres, do kterého bude výsledná obalová zóna uložena je nabízen uživateli prostřednictvím comboboxu, obsahujícího seznam editovatelných výkresů, doplněného o položku *<nový výkres>*¹. Hlavní dialog je sice dialogem modálním, nicméně v případě výběru kreslicího klíče a výplně plochy identifikací nebo při metodě zadání dat pro tvorbu obalové zóny oblastí či identifikací je dialog ukončen a vstup požadovaných dat je realizován *inputem* z grafiky, tzn. prostředím nemedálním. Vypnutí editovatelnosti během inputu zapříčiní to, že po navrácení do dialogu není výkres

¹Založení nového výkresu proběhne až po stisknutí tlačítka *OK*.

nabízen v comboboxu. Na funkci to však nemá vliv, vždy je k dispozici již zmiňovaný <nový výkres>. Pokud je během inputu vypnuto kreslení výkresu nebo jeho identifikovatelnost, funkce to zaznamená a při výběru prvku nahlásí "žádný prvek nenalezen". Na zavření grafického okna funkce nijak nereaguje a čeká, až bude opět nějaké okno dostupné.

Zadání vrstvy je řešeno editlistem, který obsahuje seznam vrstev všech otevřených výkresů, narozdíl od comboboxu umožňuje zadat i něco jiného, v našem případě vlastní název vrstvy. Pokud by vrstva nebyla zadána, bude uživatel po stisku tlačítka *OK* upozorněn a funkce bude čekat na zadání vrstvy. Podoba obou prvků je znatelná z obrázku 3.1.



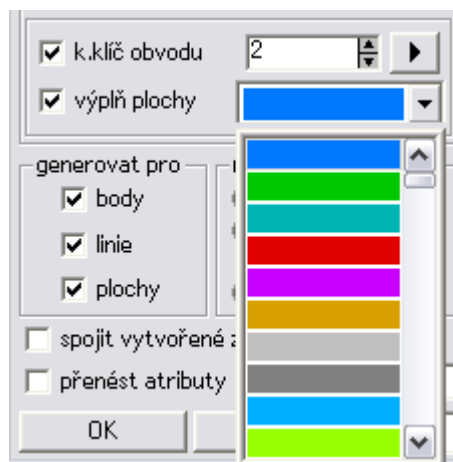
Obr. 3.1: Nastavení jednoduché zóny

3.2 Jednoduchá zóna

Při zvolení jednoduché zóny je nutno zadat její šířku, a to buď hodnotou v metrech pomocí editačního pole nebo pomocí editlistu, ve kterém je seznam všech atributů

otevřených výkresů. V případě zadání šířky hodnotou bude výsledná obalová zóna od zvolených prvků vždy o zadanou vzdálenost, naopak v případě zadání šířky atributem tomu tak být nemusí. Příkladem může být atribut šířky koruny stromu, která u každého stromu může být různá, a tudíž při zvolení tohoto atributu a výběru stromů bude mít obalová zóna stromu šířky jiné. Editační pole pro šířku hodnotou je sice typu edit, tzn. lze zadat libovolný text, nicméně zadávaná hodnota je testována. Pokud je hodnota číslo, je editační pole naplněno, v opačném případě se tak nestane. Při zadávání hodnoty do editu či editlistu je příslušný radiobox automaticky přepnut. Pro tvořenou obalovou zónu je potřeba dále zadat do políčka editintu kreslicí klíč obvodu zóny nebo kód výplně plochy anebo obojí. Jaká z hodnot bude vzata do zpracování je vyznačeno checkboxem. Při změně jedné z hodnot je příslušný checkbox automaticky zaškrtnut. Vzhled prvků je vidět na obrázku 3.1.

Hodnoty lze zadat několika způsoby. První způsob je zadáním hodnot z klávesnice, další pomocí šipek editrealu. Posledním je klik na button, kde je připravena místní nabídka. Zde lze vybrat volbu identifikace klíče či výplně plochy v grafice, nebo výběr klíče či výplně plochy z nabídky. Výběr z nabídky je možný pouze v případě, když je vybrán konkrétní výkres pro tvořenou zónu, protože u zatím neexistujícího nového výkresu není předem znám vzhled čar a výplní ploch. Teoreticky by bylo možné pro nový výkres nabídnout implicitní vzhled čar a výplní ploch, ale zvolil jsem postup podle obdobných funkcí systému MISYS. Nabídka kreslicího klíče obvodu nebo výplně plochy se zobrazí přes editreal a button (obr. 3.2).

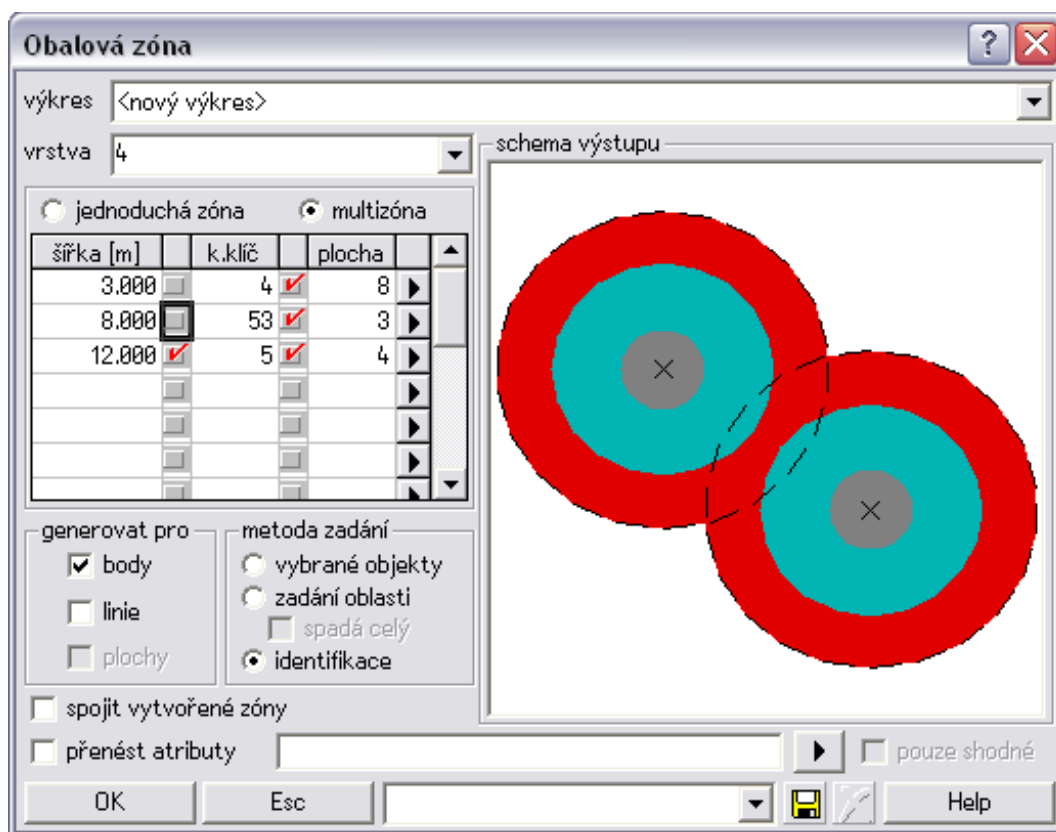


Obr. 3.2: Jednoduchá zóna - nabídka výplně ploch

Při volbě identifikace v grafice je dialog ukončen a lze vybrat přes input kreslicí klíč či výplň plochy, jejichž číslo bude převzato. Pokud uživatel při dotazu na identifikaci kreslicího klíče nebo výplně plochy stiskne tlačítko *Esc*, funkce se vrátí do dialogu, kde zůstane vyplněna původní hodnota.

3.3 Multizóna

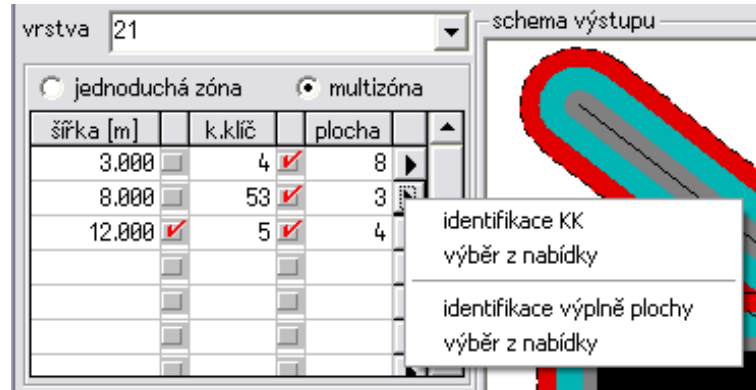
Pokud umíme systémově vyřešit problematiku jedné zóny, nabízí se vyřešit více zón najednou (obr. 3.3). Tato myšlenka je složitější na realizaci uživatelského rozhraní (jak nastavení více zón vhodně zadat), než na realizaci samotné výkonné funkce.



Obr. 3.3: Nastavení více zón

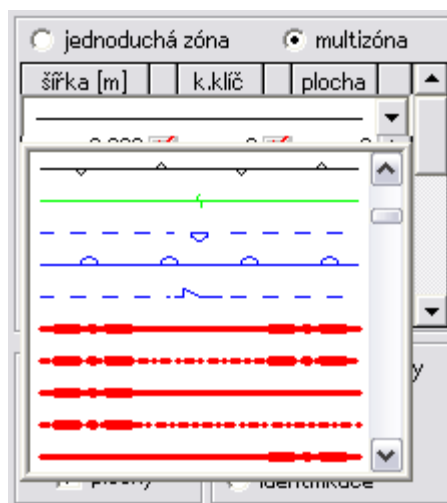
Zadání dat pro více zón je řešeno pomocí tabulky, která obsahuje sloupec typu real pro zadání šířek zón, dále sloupce typu int pro kreslicí klíč a výplň plochy. Navíc u těchto dvou sloupců je vždy sloupec s checkboxem pro zaškrtnutí, zda klíč nebo výplň plochy má být uvažován při tvorbě zóny. Pro zadané hodnoty platí stejná

kritéria jako u jednoduché zóny, multizóna však neumožňuje zadat šířku atributem, navíc volby pro identifikaci a výběr nabídkou jsou pro klíč i výplň v jedné nabídce, která je k dispozici opět po kliku na šipku na konci řádku tabulky (obr. 3.4).



Obr. 3.4: Nabídka pro kreslicí klíč a plochu

Výběr z nabídky je řešen stejně jako v případě jedné zóny, nicméně nabídka zabírá celý řádek tabulky. Ukázka nabídky po rozkliknutí je na obrázku 3.5.



Obr. 3.5: Výběr kreslicího klíče nabídkou

Zadané hodnoty v tabulce jsou při kliku na tlačítko *OK* kontrolovány. Pokud by byly například totožné šířky nebo chyběl nějaký z údajů, je uživatel upozorněn a dialog čeká na zadání potřebných hodnot.

3.4 Generovat pro

Obalovou zónu lze vytvářet k bodovým, liniovým nebo plošným prvkům, stejně tak i k jejich vzájemným kombinacím. Funkce při svém spuštění zjistí, jaká data kreslené výkresy obsahují a podle toho checkboxy pro volbu prvků buď uživateli zpřístupní, nebo zakáže. Pokud by při stisku tlačítka *OK* nebyl vybrán ani jeden prvek pro tvobu zóny, bude uživatel upozorněn a dialog vyčká, dokud nebude prvek zadán.

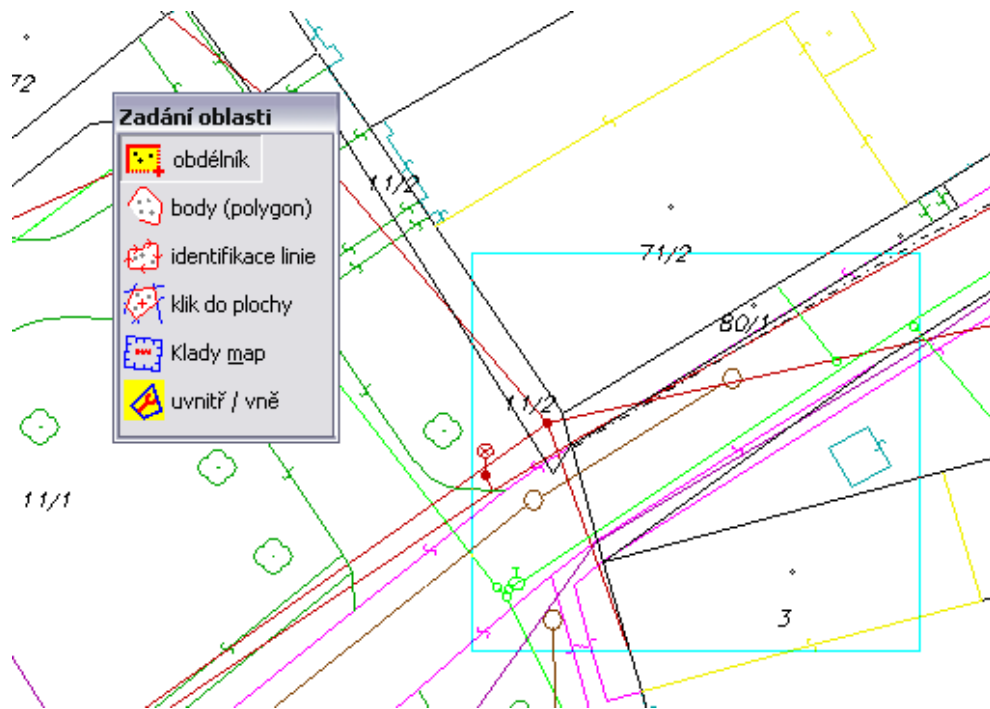
V tomto ohledu se funkce liší od obdobných funkcí z produkce ESRI nebo jiných GIS pracujících s daty ve formátu SHP. Tento formát neumožňuje v jedné vrstvě kombinovat body, linie a plochy, a proto samotná volba vrstvy určí, pro jaký typ prvků zónu generovat. Pokud některý z produktů umí generovat zónu pro více vstupních vrstev, vytvoří se zóna pro typy prvků dle vybraných vrstev, tj. i pro kombinace bodů, linií a ploch. Systém MISYS obecně nemusí mít v jedné vrstvě pouze jeden typ prvků, a tak jsem funkci doplnil o možnost omezit vstup na určitý typ prvků.

3.5 Metoda zadání

V systému MISYS může být otevřen libovolný počet výkresů. Proto bylo potřeba vymyslet co nejvhodnější způsob, jak vstupní data zadat. Jednou z možností by mohl být vstup konkrétního výkresu, tento způsob by však byl ve výsledku velmi omezující. Vstupní data lze mimo to zadat třemi jinými způsoby. První volbou jsou *vybrané objekty*. Vybranými objekty jsou takové objekty, které se kreslí v aktuálním grafickém okně. V systému MISYS jsou tomu nejčastěji *Projekty* v podobě zapnutých *Témat*, protože ty jsou asi hlavním způsobem při práci s rozsáhlejšími daty (např. ÚAP). Do zpracování pak vstoupí právě viditelné objekty.

Druhým způsobem je zadat data *oblastí*. V tomto případě je po stisku tlačítka *OK* dialog ukončen a je požadováno zadání oblasti, pro kterou se má zóna vypracovat. Pokud není zaškrtnutá volba *spadá celý*, jsou vybrány všechny objekty, které do oblasti zasahují alespoň částečně. Oblast lze zadat obdélníkem, polygonem, identifikací linie, klikem do plochy nebo mapovým listem. Pro všechny volby je možné zvolit, zda se mají brát objekty uvnitř či vně (obr. 3.6). Po zadání oblasti funkce vytvoří obalovou

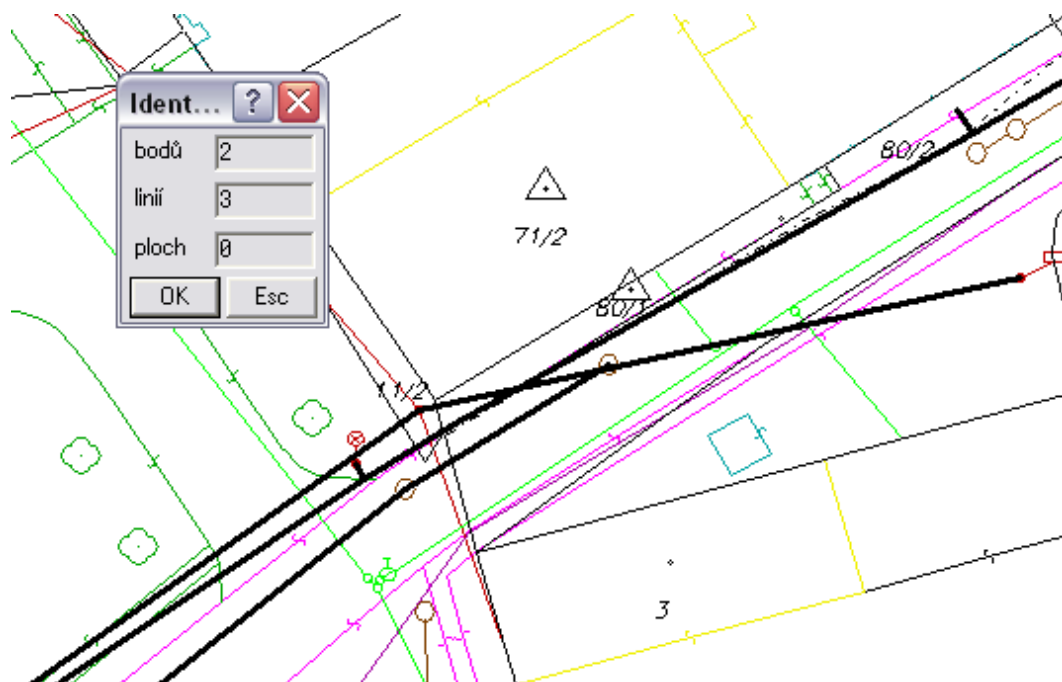
zónu a skončí. V případě stisku klávesy *Esc*, bude zadání oblasti přerušeno a funkce nabídne zpět hlavní dialog.



Obr. 3.6: Výběr dat zadáním oblasti

Poslední možností je zadat data *identifikací*. V takovém případě je po stisku tlačítka *OK* opět dialog ukončen a je nabídnut nemoďální dialog propojený s inputem. Dialog zobrazuje počet identifikovaných objektů zvlášť pro body, linie i plochy. Pokud v hlavním dialogu bude zvoleno generovat zónu například pouze k bodům, bude dialog pro identifikaci upraven (zobrazován bude jenom počet identifikovaných bodů). Pro vytvoření obalové zóny k identifikovaným objektům je zapotřebí stisku tlačítka *OK*, při stisku *Esc* se ukončí identifikace a je zobrazen zpět hlavní dialog. Vzhled funkce při identifikaci je názorný z obrázku 3.7, kde byly identifikovány dva body vyznačené symbolem v podobě trojúhelníku a tři linie vyznačené tlustou černou čarou. V obou posledních případech zadání dat, stejně tak i při identifikaci kreslicího klíče a výplně plochy by mohlo dojít k zavření stávajících výkresů nebo k otevření výkresů nových. V takové situaci funkce zareaguje odebráním či přidáním výkresu do nabídky comboboxu, nicméně nabízené vrstvy či atributy se v dialogu

nemění².



Obr. 3.7: Výběr dat identifikací

3.6 Schéma výstupu

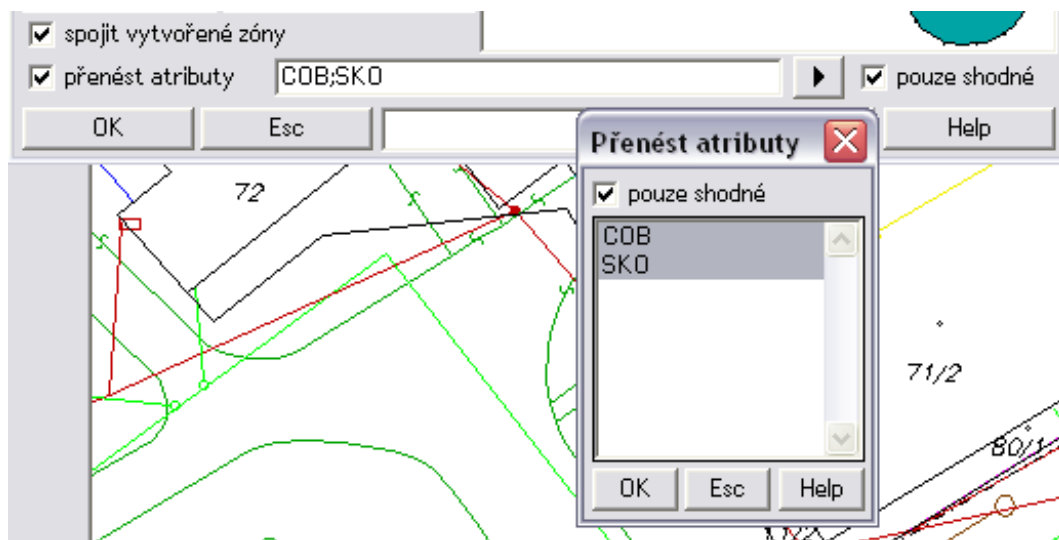
Schéma je realizováno pomocí grafického okna vloženého do dialogu. Toto okno má úmyslně zakázané posouvání, identifikaci v něm atd., jediné co je možné s oknem provádět je *zoomování* prostředním tlačítkem myši, které umožňuje si kresbu v určitém místě zvětšit. Cílem schématu je co nejvíce uživateli přiblížit, jaké má nastavení dialogu a jak bude výsledná obalová zóna vypadat. Kresba tedy zohledňuje, pro jaký typ dat se bude zóna generovat, jakou bude mít zóna výplň a kreslicí klíč obvodu, dále zda tvořené zóny spojit či nikoli. U více zón je ještě kvůli názornosti zohledňován poměr hodnot šířek. Skutečná šířka není tedy ani u jednoho typu zóny v kresbě zohledněna. Důvod je ten, že v grafickém okně nejsou zobrazeny reálná data (body, linie, plochy), a tudíž při zadání nevhodné šířky by bylo schéma nečitelné. Pokud by nebylo zvoleno generování ani k jednomu typu dat, schéma zůstane bez kresby.

²Odebírání a přidávání výkresů během chodu funkce se nepředpokládá a nedoporučuje se.

Jak je z obrázku 3.1 a 3.3 patrné, kresba není tvořena k reálným datům. Zvoleny byly pro názornost dva body, dvě linie a dvě plochy. Pro kreslení zóny kolem těchto prvků bylo zapotřebí vypočítat body obalové zóny už v dialogu, nebylo však zapotřebí uvažovat problematiku atributů či skutečnou šířku zóny.

3.7 Spojit zóny a přenést atributy

U vytvářených zón lze navolit ještě některé další vlastnosti. Jednou z nich je, zda *vytvářené zóny spojit*. V takovém případě se vypočtou průsečíky překrývajících se zón a kresba obvodové linie zóny bude vedena tak, aby nedošlo k překrytí zón. Názorně je princip vidět na schématu v pravé části dialogu. Další možností je volba, zda k zónám *přenášet atributy* původních objektů. Pokud by nebyl checkbox pro přenášení atributů zapnut, žádný z atributů se nepřenesou. V případě jeho zapnutí společně s volbou sloučení zón je možné přenést *pouze shodné* atributy. Pro zadání přenosu jiného z atributů slouží editační pole³ nebo tlačítko s šipkou, které vyvolá dialog pro snadnější výběr požadovaných atributů. Vzhled prvků a dialog pro výběr atributů je uveden na obrázku 3.8.



Obr. 3.8: Dialog pro výběr přenášených atributů

³Atributy lze oddělit mezerou, čárkou nebo znakem "|".

3.8 Poslední řádek

Mezi poslední prvky dialogu patří tlačítko *OK*, po jehož stisku buď dojde k vytvoření obalové zóny kolem vybraných prvků, nebo bude požadováno zadání dat oblastí či identifikací, popřípadě bude požadováno zadání nového výkresu. Toto tlačítko však také kontroluje správnost zadaných hodnot v dialogu. Pokud jsou hodnoty správné, dojde k jednomu z uvedených případů a k uložení nastavení dialogu do inicializačního souboru. Zápis nastavení v souboru je následující:

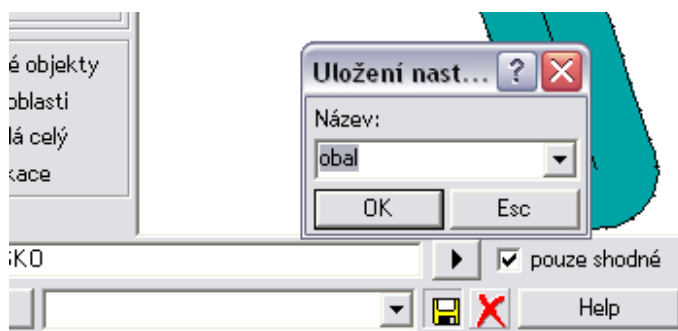
[ZLBUFFER]	'název sekce inicializačního souboru
SPOL=1 159 svd 0	'společné nastavení pro jednoduchou zónu i více zón
SINGLE=0 1 1 2 1 3	'nastavení pouze pro jednoduchou zónu
MULTI=3 3.00 -4 8 8.00 85 3 12.00 5 4	'nastavení pouze pro více zón
vyk=C:\wkokes953\DEMO\LIP_STAN.VYK	'výkres pro výstup
saves=;c;d	'seznam názvů uložených nastavení dialogu
A0=21 135 0 1 1 2 1 3	'hodnoty uloženého nastavení dialogu uživatel
A1=21 135 0 1 3 er 1 2 1 3	'hodnoty uloženého nastavení dialogu uživatel

Pokud však nejsou hodnoty korektní, bude uživatel upozorněn následujícími hlášeními:

- Nebyla zvolena žádná cílová vrstva.
- Není zadána šířka obalové zóny.
- Chybný kreslicí klíč.
- Nebyl vybrán obvod ani výplň plochy.
- Zadány totožné šířky.
- Nebyla zvolena žádná akce.

Tlačítko *Esc* ukončí dialog bez jakéhokoli dotazu či varování. K funkci byla napsána i nápověda, jejíž znění bude vyvoláno po stisku tlačítka *Help*. Ačkoli pro běh funkce není nápověda nutná, je její důležitou součástí, a proto by neměla být opomíjena. Nápověda slouží především pro ty, co funkci pouští poprvé nebo po delší době a chtějí se dozvědět resp. zopakovat, jak se funkce ovládá a co případně dělá.

Poslední možností dialogu je uložení jeho libovolného nastavení pod určitým názvem. Kdykoli v budoucnu může uživatel vybrat název příslušného nastavení a dialog bude nastaven do takového stavu, ve kterém byl pod názvem uložen. Ukládané hodnoty se ukládají do stejného souboru a sekce jako v případě ukládání po stisku *OK*, což je znatelné ze stejné ukázky, kde uloženým nastavením odpovídá klíč *saves*, *A0* a *A1*. Uložené nastavení je samozřejmě možné i smazat. Ukázka vzhledu prvků pro ukládání nastavení je na obrázku 3.9.



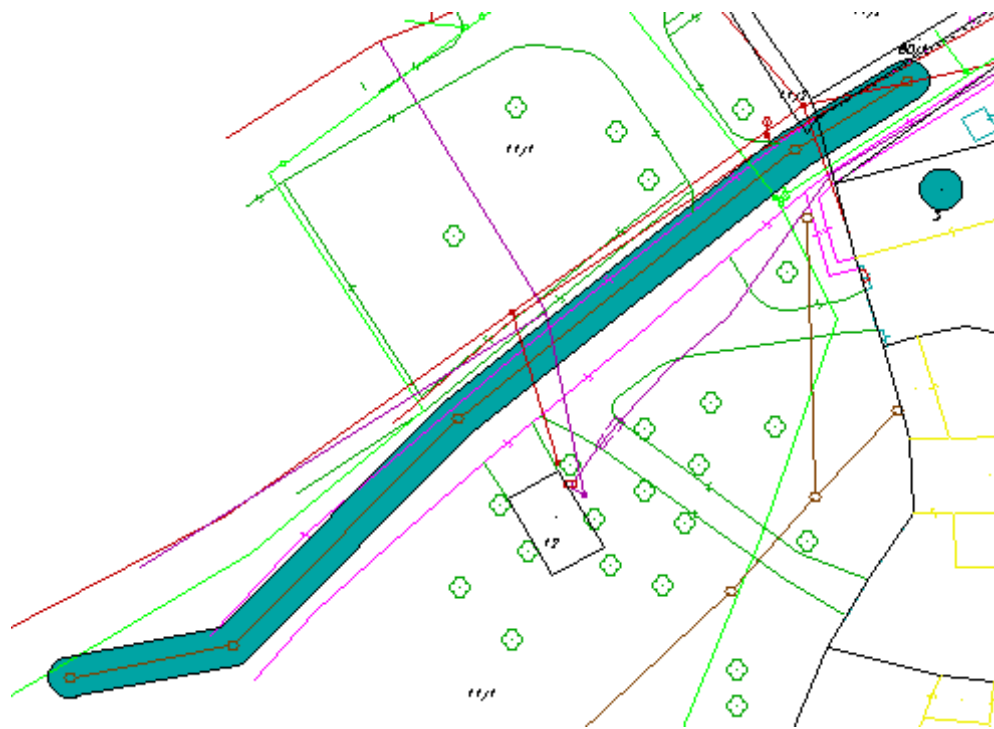
Obr. 3.9: Dialog pro uložení aktuálního nastavení

Velikost dialogu lze měnit pouze roztažením směrem doleva nebo doprava. Maximální rozměr roztažení není omezen, narozdíl od minimálního rozměru, který je nastaven na šířku uvedenou v definici dialogu. Při změně šířky jsou roztahována všechna editační pole (což je vhodné v případě, když se text do pole nevejde celý), ostatní prvky dialogu jsou podle potřeby posouvány.

3.9 Výsledky obalových zón

Závěrem kapitoly bych rád uvedl výsledek tvorby jednoduché obalové zóny (obr. 3.10) a výsledek při volbě více zón (obr. 3.11). Na prvním obrázku byla tvořena zóna k jedné linii a bodu, na druhém byla tvořena k ploše, linii a dvěma bodům. Jak je na obrázku vidět, nebylo zvoleno spojení zón, proto jsou přes sebe kresleny obvodové linie obou prvků.

Funkce je od října 2009 součástí systému (verze MISYS 9.52). Dostupná je z položky menu *Výkres/Analytické funkce/Obalová zóna*.



Obr. 3.10: Výsledek jednoduché obalové zóny



Obr. 3.11: Výsledek obalové zóny při volbě více zón

4 Atributová tabulka

Atributová tabulka (obr. 4.1) je v našem případě tabulka objektů a jejich atributů vztažených ke konkrétnímu výkresu. Co řádek tabulky to jeden objekt, co sloupec to název atributu¹. Kromě zobrazování prostého seznamu hodnot atributů vztažených k objektu je tabulka spjata úzce s grafikou a obsahuje další tlačítka a ikony, např. pro vyhledávání v tabulce či její export do jiného formátu.

Tabulka umožňuje jak prohlížení, tak editaci atributů (není potřeba se mezi těmito způsoby přepínat). Na obsah tabulky se nelze dotazovat pomocí některého dotazovacího jazyku. Hlavní dialog funkce je dialogem nemodálním, což přináší mnohé výhody pro uživatele a mnohé starosti pro programátora. V první podkapitole je zmíněn vznik tabulky společně s jejím plněním a popis atributů. Obsahem další podkapitoly je význam a popis funkčnosti ikon a tlačítek dialogu.

The screenshot shows a dialog box titled 'Tabulka atributů: C:\zdenek\cvut\diplomka\uap\UAP_trunk\Podklady...'. The dialog contains a table with the following data:

objekt	vrstva	DATUM	JMENO	PLATNOS	TYP
1	UAP_COV_BOD	16.07.2008	ČOV Broumy	31.07.2008	K = komunální
3	UAP_COV_BOD	16.07.2008	ČOV Hředle	31.07.2008	P = průmyslové
4	UAP_COV_BOD	16.07.2008	ČOV Chrástany	31.07.2008	K = komunální
17	UAP_COV_BOD	16.07.2008	ČOV Jesenice	31.07.2008	S = smíšená
5	UAP_COV_BOD	16.07.2008	ČOV Kněževés	31.07.2008	K = komunální
6	UAP_COV_BOD	16.07.2008	ČOV Kolečovice	31.07.2008	K = komunální
12	UAP_COV_BOD	16.07.2008	ČOV Křivoklát	31.07.2008	K = komunální
18	UAP_COV_BOD	16.07.2008	ČOV Lišany	31.07.2008	K = komunální
16	UAP_COV_BOD	16.07.2008	ČOV Lubná	31.07.2008	K = komunální
7	UAP_COV_BOD	16.07.2008	ČOV Mutějovice	31.07.2008	K = komunální
19	UAP_COV_BOD	16.07.2008	ČOV Nové Strašec	31.07.2008	S = smíšená
14	UAP_COV_BOD	16.07.2008	ČOV Pavlíkov	31.07.2008	K = komunální
15	UAP_COV_BOD	16.07.2008	ČOV Rakovník	31.07.2008	P = průmyslové
10	UAP_COV_BOD	16.07.2008	ČOV Rakovník	31.07.2008	K = komunální
13	UAP_COV_BOD	16.07.2008	ČOV Rostoky	31.07.2008	K = komunální
8	UAP_COV_BOD	16.07.2008	ČOV Ruda	31.07.2008	K = komunální

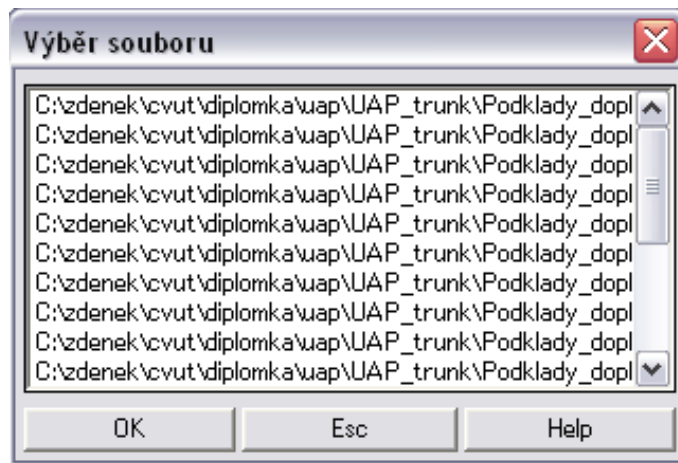
At the bottom of the dialog, there are three buttons: 'Esc', 'Export', and 'Help'.

Obr. 4.1: Hlavní dialog atributové tabulky

¹Vyjímkou je sloupec s čísly objektů a sloupec s názvy vrstev.

4.1 Vznik tabulky a její plnění

Pro spuštění funkce je požadován alespoň jeden kreslený výkres. Pokud takový výkres není, je uživatel upozorněn příslušným hlášením v protokolu. V případě pouze jednoho výkresu je spuštěna přímo atributová tabulka, v opačném případě je nabízen dialog se seznamem výkresů (obr. 4.2). Výkres lze vybrat buď z nabízeného seznamu nebo pomocí kliku do grafiky, pokud je příslušný výkres zapnutý pro identifikaci.



Obr. 4.2: Dialog pro výběr výkresu

Pozn. Za běhu atributové tabulky lze měnit výkresy, jejichž atributy mají být v tabulce prezentovány. Změna výkresu je realizována stejným způsobem jako výběr výkresu při spuštění funkce.

Tabulky v systému MISYS a jejich sloupce s uvedením jejich typu jsou definovány samostatně mimo definici dialogu a do dialogu jsou poté vkládány. U většiny tabulek je znám předem počet sloupců a jejich typ, a proto mohou být nadefinovány dopředu. U atributové tabulky tomu tak není, jelikož počet sloupců a jejich typ získáme až po vzniku dialogu analýzou objektů ve výkresu. Proto je tabulka na začátku definována pouze s jedním sloupcem, je vložena do definice dialogu, a až při vzniku dialogu je plněna dalšími sloupci. Tento způsob s sebou přináší problematičtější přístup k buňkám tabulky při programování, nicméně v tomto případě je jeho použití nezbytné.

Výkresy v systému MISYS mohou obsahovat dva typy atributů:

Technologické atributy jsou definovány v souboru definic atributů SAT svým názvem, typem, délkou (velikostí), údajem o povinnosti atribut zadat a případně číselníkem hodnot. Soubor definice technologických atributů se k výkresu připojí pomocí technologie tvorby výkresu (viz **Nastavení programu/Technologie**).

Netechnologické atributy jsou všechny ostatní atributy. Jsou vždy typu řetězec a nejsou povinné.

Při vzniku dialogu je první sloupec naplněn čísly od 0 do $n-1$ (n se rovná počet objektů výkresu), které se používají jako index do pole s popisovači objektů pro následné vysvícení vybraných objektů (řádků) tabulky v grafice. Tento sloupec je jako jediný skrytý a není tedy možné k němu uživatelsky přistoupit. Druhý sloupec je naplněn čísly (identifikátory) objektů, podle kterých uživatel pozná, s kterými objekty pracuje. Teoreticky pro vysvícení objektů by stačil sloupec s čísly objektů, ale jelikož se dají řádky tabulky třídit, je programátorsky pohodlnější si vyzvednout index přímo do pole s popisovači než převádět číslo objektu na potřebný index. Jelikož číslo objektu je přiděleno automaticky a nelze ho uživatelsky měnit (nemá tedy cenu jej editovat v tabulce). Třetí sloupec obsahuje název vrstvy. Všechny tyto tři sloupce nejsou atributy.

Další sloupce už jsou plněny jenom atributy. Každý sloupec obsahuje v záhlaví tabulky jméno atributu, podle kterého lze poznat, zda se jedná o atribut technologický (označen tučným písmem) či nikoli. Klik na jméno atributu způsobí, že řádky tabulky budou podle toho atributu setříděny. Při přidávání sloupců do tabulky během vzniku dialogu je vyhledán nejdelší text v sloupci. Podle tohoto textu je pak nastavena šířka sloupce, tzn. že by se měl text do sloupce vejít vždy celý. Šířku sloupců je pak možné kdykoli změnit tažením pravého okraje záhlaví sloupce.

Každý atribut má svůj typ. Netechnologický atribut je vždy řetězec, technologický však může být kromě řetezce ještě typu číslo a datum nebo může mít formu číselníku hodnot (obr. 4.3). Všechny typy lze dvojklikem do tabulky editovat podle jejich typu, jediné u číselníku hodnot lze vybrat pouze z předdefinované nabídky.

PLATNOS	TYP	▲
31.07.2008	K = komunální	▼
31.07.2008	P = průmyslová	▼
31.07.2008	K = komunální	▼
31.07.2008	K = komunální	▼
31.07.2008	P = průmyslová	▼
31.07.2008	S = smíšená	▼
31.07.2008	K = komunální	▼
31.07.2008	K = komunální	▼
31.07.2008	K = komunální	▼

Obr. 4.3: Ukázka technologického atributu s číselníkem hodnot

Typ atributu určuje, jaký bude mít hodnota atributu tvar. Za řetězec je považován libovolný textový zápis² o konečné délce. U čísla lze definovat, kolik bude mít desetinných míst a jakou bude mít délku, u hodnoty typu datum je délka vždy 10 znaků. Při editaci buňky typu datum lze vyvolat kalendář (obr. 4.4) pro výběr požadovaného data. Pro definici technologických atributů slouží funkce *Výkres/Atributy objektů/Definice technologických atributů*.

objekt	vrstva	DATUM	JMENO	PLATNOS	TYP	▲
1	UAP_COV_BOD	16.07.2008	ČOV Broumov	31.07.2008	K = komunální	▼
				31.07.2008	P = průmyslová	▼
				31.07.2008	K = komunální	▼
				31.07.2008	S = smíšená	▼
				31.07.2008	K = komunální	▼
				31.07.2008	K = komunální	▼
				31.07.2008	K = komunální	▼
				31.07.2008	K = komunální	▼
				31.07.2008	K = komunální	▼
				31.07.2008	K = komunální	▼
				31.07.2008	P = průmyslová	▼
				31.07.2008	K = komunální	▼
8	UAP_COV_BOD	16.07.2008	ČOV Ruda	31.07.2008	K = komunální	▼

1. August 2008						
August 2008						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Obr. 4.4: Editace technologického atributu typu datum

²Může obsahovat i číslice a libovolné znaky.

Tabulka však nemá sloužit pouze k editaci, ale i k prohlížení či vyhledávání objektů s jejich atributy. Prohlížení je možné oboustranně, a to jak klikem do tabulky a následným zvýrazněním v grafice, tak i klikem na objekt v grafice a následným označením příslušného řádku v tabulce. Při výběru v grafice je však možné vybírat objekty pouze po jednom, v tabulce je však možné vybrat více řádků najednou, a tudíž zvýraznit více objektů. Tabulka tedy rozlišuje klik a dvojklik na příslušný řádek. Při kliku je celý řádek označen černě. Výběr bloku je realizován buď tažením myši nebo standardně pomocí klávesy *Shift*. Jednotlivé vybírání více řádků je možné pomocí klávesy *Ctrl*, jak jsme tomu zvyklí z jiných aplikací.

4.2 Funkčnost ikon a tlačítek

Horní část dialogu obsahuje tlačítka s ikonami pro obsluhu tabulky. Některé ikony byly převzaty ze systému MISYS, většina z nich však byla nově vytvořena speciálně pro tuto funkci. Ikony se načítají do paměti při prvním spuštění dialogu. Použití ikon na tlačítka dialogu se zdá být nejvhodnějším způsobem, jak problematiku ovládní vyřešit. Při nájezdu myši nad ikonu je zobrazen krátký *tooltip*, obsahující text s názvem tlačítka. V následujícím výčtu se pokusím popsat funkčnost jednotlivých ikon.

.....



Pokud je některý řádek či více řádků v tabulce vybráno, budou tyto řádky po zmáčknutí ikony seřazeny na začátek tabulky. Seřazení nemá vliv na zvýrazněné objekty v grafice. Pokud nebude označen žádný řádek nebo budou naopak označeny všechny, seřazení se neprovede. Pokud byly řádky podle nějakého atributu setříděny klikem na jeho záhlaví, bude toto setřídění zrušeno.

.....



V případě, že nejsou všechny řádky tabulky označeny, dojde k jejich výběru. Tím je docíleno, že jsou vysvíceny všechny objekty výkresu, pro který je tabulka vytvořena.



Jestliže některý z řádků je vybrán, dojde k jeho odznačení. Všechny vysvícené objekty ve výkresu jsou zhasnuty.



Jedná se o opak aktuálního výběru v tabulce, tzn. že vybrané řádky tabulky jsou odznačeny a ostatní jsou označeny. Pokud tedy není označen žádný řádek, jsou označeny všechny a naopak.



Po zmáčknutí tlačítka je vyvolán modální dialog (obr. 4.5) pro zadání nového názvu netechnologického atributu³. Nelze tedy zvolit typ atributu, hodnoty budou považovány za textový řetězec. Po zmáčknutí tlačítka *OK* je na konec tabulky přidán nový prázdný sloupec. Kdyby byl název atributu zadán uživatelem malými písmeny, bude vložen do tabulky s velkými (název atributu má být v systému MISYS velkými písmeny). Dále je na tlačítku *OK* testováno, zda některý ze sloupců tabulky už zadaný název neobsahuje. Pokud by obsahoval, byl by uživatel upozorněn a sloupec by nebyl vložen.



Obr. 4.5: Dialog pro zadání nového atributu

³Technologické atributy nelze pomocí atributové tabulky vytvářet.



Po stisku je vyvolán modální dialog (obr. 4.6), který obsahuje názvy atributů. Stejně jako v případě vkládání nových atributů nelze technologické atributy mazat, proto nejsou ani v listboxu dialogu nabízeny. Lze vybrat, a tudíž vymazat více atributů najednou. Nelze mazat první tři sloupce, tzn. indexy do pole, čísla objektů a vrstvy.



Obr. 4.6: Dialog pro výběr netechnologického atributu pro smazání



Poslední volbou na ovládání sloupců je jejich skrývání. Po stisku ikony je opět vyvolán modální dialog (obr. 4.7), jehož hlavním prvkem je tabulka obsahující všechny atributy a navíc i sloupec s vrstvou. Zda atribut skrýt či naopak zobrazit určuje stav checkboxu u každého z názvů. Zaškrtnutí znamená, že má být atribut v tabulce viditelný. Tažením myši přes názvy atributů lze provést vícenásobný výběr⁴. Skrytí nebo zobrazení pouze zohledňuje viditelnost sloupců, jejich počet se tím nemění.

⁴Zda atributy vybrat nebo odznačit určuje stav prvního checkboxu.

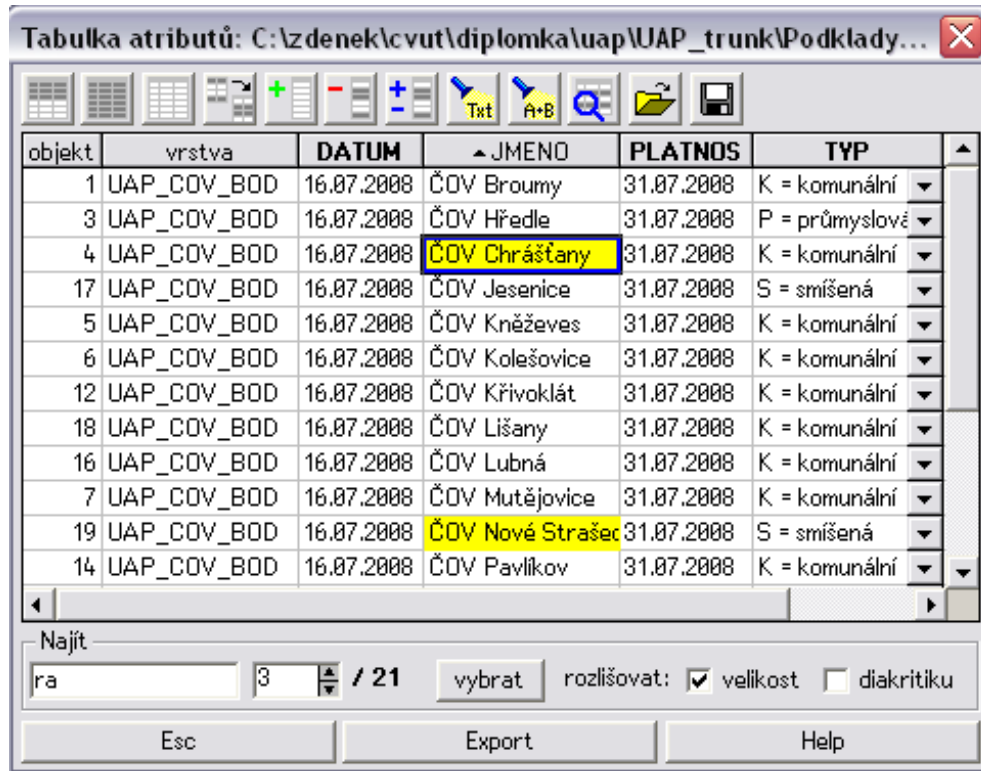


Obr. 4.7: Dialog pro skrývání atributů



Dialog je po stisku ikony rozšířen o další prvky pro hledání, které jsou zobrazeny v řádku těsně pod tabulkou (obr. 4.8). Rozměr dialogu je tedy za běhu zvětšen a tlačítka dialogu jsou odsunuta níže. Při opětovném stisku ikony dojde naopak k zmenšení dialogu, který dostane původní rozměr.

Mezi nové prvky patří editační pole pro zadání hledaného textu, dále editint pro posouvání v buňkách, kde byl požadovaný text nalezen. Uživatel je informován o počtu nalezených buněk, které jsou vybarveny žlutým pozadím. To znamená, že je vyhledáváno, zda se text nevyskytuje aspoň jednou v některé z buněk. Pokud by hledaný řetězec byl v jedné buňce vícekrát, nebude to v počtu nalezených výskytů zohledněno. K dalším možnostem patří tlačítko *Vybrat*, po jehož stisku budou řádky s výskytem hledaného textu označeny černě - budou vysvíceny v grafice a je možné je např. setřídit na začátek tabulky. Při hledání lze nastavit, zda *rozlišovat velikost* písmen a zda uvažovat *diakritiku*. Pokud stav některého z checkboxů bude během hledání změněn, dojde k novému vyhledání, stejně tak tomu je při změně textu

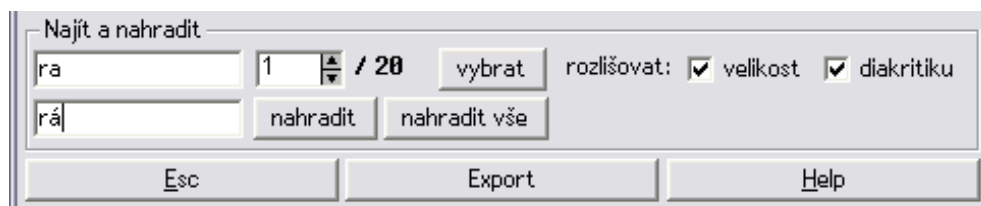


Obr. 4.8: Ukázka dialogu při zapnutém hledání

v editačním poli. Pokud uživatel klikne na řádek tabulky, je hledaný text v editačním poli smazán a žluté podbarvení zmizí. Jako jediný není do vyhledávání zahrnut sloupec s indexy do pole pro zvýraznění objektů a sloupce, které jsou zrovna skryty.



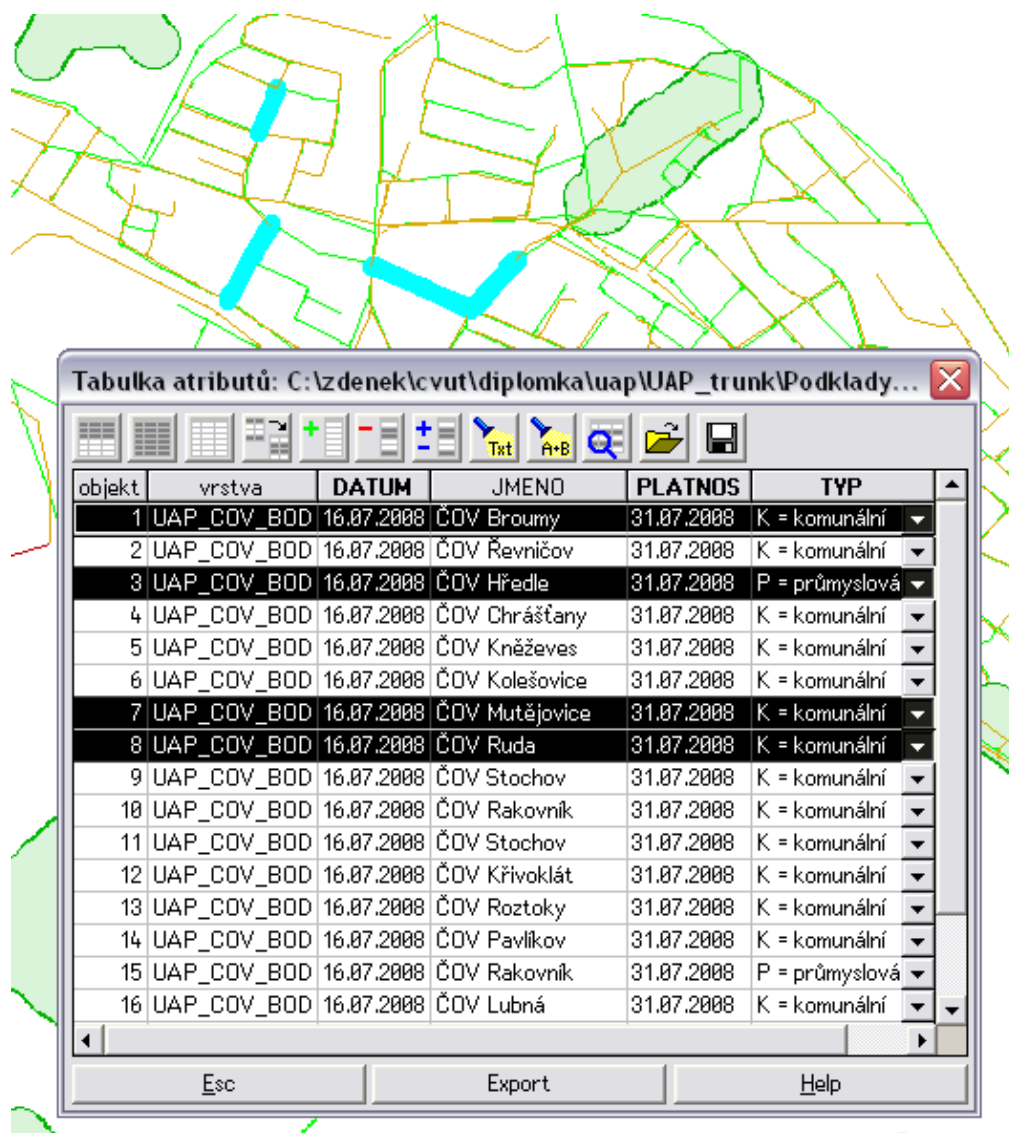
Dialog je též rozšířen o prvky popsané u vyhledání a je doplněn o editační pole pro text, za který má být nalezený text nahrazen, dále o tlačítka *nahradit* a *nahradit vše* (obr. 4.9). Ačkoli je zvýrazněna celá buňka, je vždy nahrazen pouze text, který byl vyhledáván. Volba *nahradit* přepíše pouze text v buňce, která je aktivní.



Obr. 4.9: Řešení nahrazování v tabulce



Objekty reprezentované jedním řádkem tabulky lze pomocí této funkce zobrazit ve vhodném výřezu do grafického okna. Při výběru jednoho řádku je zobrazen opsaný obdélník pouze vybraného objektu, při více řádcích je počítán opsaný obdélník pro všechny vybrané objekty (obr. 4.10). Nemá tedy příliš smysl vytvářet výřez pro velké množství objektů, navíc pokud nejsou blízko sebe.



Obr. 4.10: Zoom na vybrané objekty

Z uvedeného obrázku lze usoudit, že výřez není správně - v tabulce jsou totiž označeny čtyři řádky, ale v grafice jsou vysvíceny tři objekty. Čtvrtý vysvícený objekt je však schován pod dialogem.



Po stisku ikony je hlavní dialog s tabulkou ukončen. Pokud byla v tabulce provedena změna dat, je uživatel dotazován, zda obsah uložit či nikoli. V další fázi je nabízen dialog pro výběr jiného souboru (obr. 4.2). Při stisku tlačítka *OK* je tabulka naplněna daty z nově vybraného výkresu, při *Esc* je opět nabídnuta tabulka s původními atributy.



Ikona slouží k uložení změn v tabulce. Po stisku této ikony lze ukončit tabulku, aniž by byl uživatel dotazován, zda data ukládat.

Dialog pod tabulkou obsahuje tři tlačítka. Při stisku tlačítka *Esc* je provedena kontrola, zda byla data změněna. Pokud byla, je zobrazen dotaz zda ukládat, v opačném případě je funkce ukončena. Tlačítko *Export* vyvolá standardní dialog pro uložení souborů. Zde je možné vybrat, zda obsah tabulky uložit do formátu MDB (databáze Microsoft Access), XLS (soubor Microsoft Excell) nebo do CSV (seznam atributů oddělený středníky). Při exportu se bere v úvahu, jaké sloupce jsou skryty, a tudíž se nemají exportovat. Exportován není též sloupec s indexy. Tlačítko *Help* slouží k vyvolání obecné nápovědy k funkci.

Jelikož je dialog nemožný, může uživatel přistupovat i k jiným funkcím systému a měnit vlastnosti výkresů. Funkce s touto možností počítá, a tudíž se zachová korektně. Pokud bude např. tabulka naplněna daty některého z výkresů a výkres bude zavřen, bude dialog ukončen. Není přesto vhodné měnit nastavení výkresu při práci s ním. Dialogu lze během chodu měnit libovolně rozměr - při jeho vzniku je ale vždy nastavena jeho minimální výška a šířka. Při změně rozměru je vždy upravován rozměr tabulky, u minimální výšky jsou zobrazeny pouze čtyři řádky tabulky, minimální šířka je nastavena podle velikosti dolních třech tlačítek tak, aby byly vždy celé vidět.

Funkce je specifická z hlediska programování v tom, že nelze obsloužit zadání

dat uživatelem a předat nastavení pomocí struktury výkonné funkci, jak tomu je v případě Obalové zóny. Ovládání funkce je tak úzce navázáno na dialog, že většina funkcí je proto součástí dialogu. Aktuálně je funkce předmětem interního testování ve společnosti GEPRO spol. s r.o. Po vypořádání připomínek by měla být doplněna do některé z dalších verzí systému MISYS. Vývoj funkce byl pozdržen objevením chyby v jádru systému, která se týkala práce s tabulkou, jejíž počet sloupců byl za běhu dialogu zvětšen. Chyba byla opravena.

5 Funkce překrytí (overlay)

Pro funkce překrytí je užíván též anglický výraz *overlay*. Jak bylo v první kapitole zmíněno, v následující podkapitole budou uvedeny tyto funkce a jejich význam, v dalších podkapitolách se budu věnovat uživatelskému rozhraní některých z nich.

5.1 Seznam funkcí

Klasicky se problém překrytí řešil překrytím dvou nebo více map na fóliích. Stejný problém se v GIS řeší pomocí základních algoritmů pracujících s dvěma nebo více informačními vrstvami, jejichž výsledkem je vytvoření nových objektů, které mají kombinace vlastností původních objektů [8]. V systému MISYS jsou data uspořádána tak, že lze u funkce Rozpuštění a Sjednocení pracovat pouze s jednou informační vrstvou. U dalších funkcí je však už také potřeba určit dvě vstupní vrstvy, aby bylo možné provést požadovanou operaci s jednou vrstvou nad vrstvou druhou. Seznam funkcí obecně spadajících mezi funkce překrytí je uveden v tab. 5.1.

český výraz	anglický výraz
Průnik	Intersect
Sjednocení	Union
Přirazení	Identify
Ořezání	Clip
Vymazání	Erase
Aktualizace	Update
Rozřezání	Split
Rozpuštění	Dissolve
Spojení	Merge
Připojení	Append

Tab. 5.1: Názvy funkcí s anglickým ekvivalentem

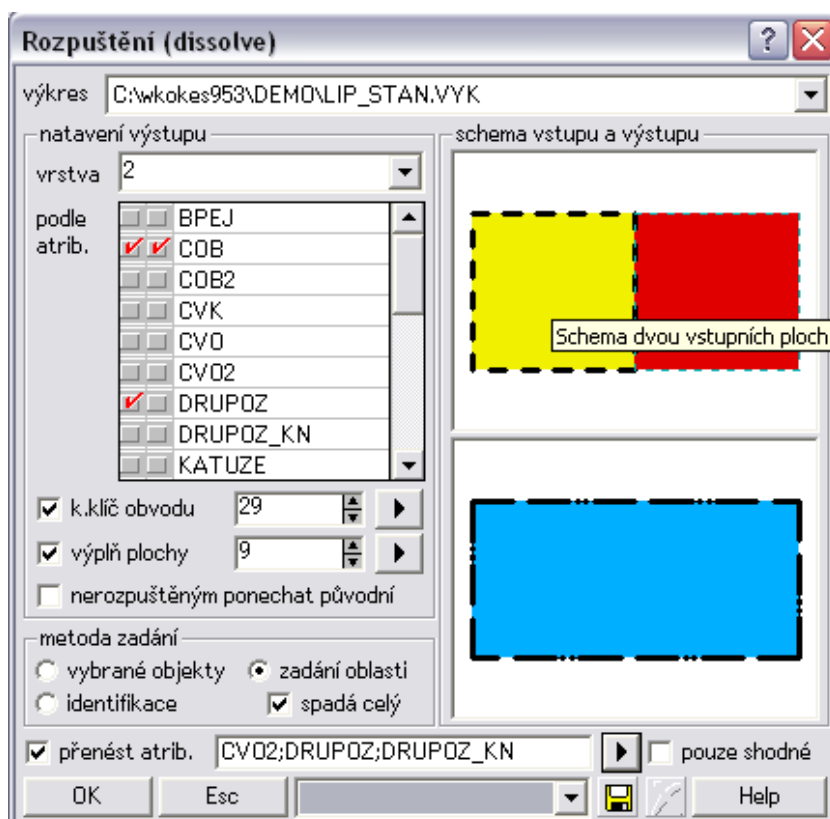
Názvy funkcí byly čerpány z článku uveřejněném v odborném časopise GeoBusiness [9]. Výsledkem funkcí jsou nové objekty, kterým jsou přiřazeny buď atributy

původní nebo nové. Tím se funkce překrytí liší od prostorových dotazů, kde žádné nové objekty nevznikají.

5.2 Funkce Rozpuštění (Dissolve)

Funkce slouží k sloučení jednotlivých plošných prvků na základě shodných atributů. To znamená, že pokud mám například přehledku katastrálních území celé ČR, budu moci jejich rozpuštěním získat kraje a vytvořit tak přehledku všech krajů v republice. Plochy, které mají být rozpuštěny, musejí mít část své obvodové linie v určitém úseku společné¹. Pokud nemají, jsou při rozpouštění ignorovány, ačkoli byly zahrnuty do výběru pro rozpuštění. Ve výsledné vrstvě se ale tyto plochy objeví.

Na obrázku 5.1 je vidět hlavní dialog zobrazený po spuštění funkce.



Obr. 5.1: Hlavní dialog funkce Rozpuštění

¹Pokud se plochy překrývají, lze pro vytvoření jejich společné obvodové linie použít funkci Sjednocení.

Díky datům, která do funkce vstupují, je tato funkce velmi podobná Obalové zóně. Tudíž i řešení uživatelského rozhraní je pojato obdobným způsobem. Proto byly některé prvky převzaty a doplněny o nové. Převzít některé chování dialogu a snažit se o podobnost vzhledu je u funkcí podobné skupiny důležité, jelikož tím se uživateli zpříjemňuje práce a usnadňuje se použitelnost funkce, především při jejím prvním spuštění. Řešení je opět rozděleno na část uživatelského rozhraní, které obslouží veškeré zadání dat uživatelem a část samotné výkonné funkce. V dalších částech kapitoly se tedy budu zabývat pouze věcmi, které jsou něčím odlišné od řešení obalové zóny.

Předání dat mezi uživatelem a výkonou funkcí je realizováno pomocí struktury, jako tomu bylo v předchozím případě. Obsah struktury je následující:

```

record dissolve_rec      'struktura
  container cnr          'cílový výkres
  string vrstva         'cílová vrstva
  string dynamic disatr 'pole s názvy atributů podle kterých se plošné prvky
                        rozpouští
  int idisatr           'délka pole
  int dynamic atrflag   'zda je atribut nutné vyžadovat pro rozpuštění u obou
                        ploch (1=vyžadován, 0=nevyžadován), délka pole stejná
                        jako u disatr
  string copy_atr       'seznam přenášných atributů oddělený středníky
  int flag              'viz DISSOLVE_*
  int klic              'hodnota kreslicího klíče (<=0 značí nekreslit obvod)
  int plocha            'hodnota výplně plochy (<0 značí nekreslit plochu)
  point dynamic pnts[0] 'body při zadání dat oblastí
  int ipnts             'počet zadaných bodů při zadání dat oblastí
  identify dynamic ide[0] 'prvky vybrané při identifikaci dat pro rozpuštění
                        (objekt, bod, typ)
  int iide              'počet vybraných objektů
end record
  
```

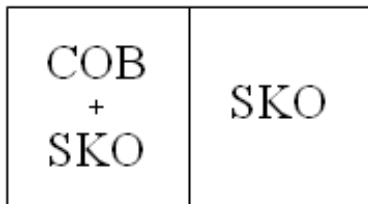
Částečně se změnil i obsah proměnné flag, která teď má tyto vlajky:

```

#define DISS_ORIG      1   'nerozpuštěným plochám ponechat původní
#define DISS_COPY_ATR  2   'přenést atributy
#define DISS_FENCE    4   'zadání oblastí
#define DISS_WHOLE    8   'spadá celý
  
```

```
#define DISS_IDENTIFY 16 'identifikace
#define DISS_OUT 32 'vně oblasti
#define DISS_IDENTICAL 64 'pouze shodné
```

Pro rozpuštění zadaných ploch je potřeba určit atribut (rozpuštění lze provést i podle více atributů), podle kterého se má operace provést. Tabulka obsahuje tři sloupce, kde třetí ze sloupců obsahuje všechny názvy atributů kreslených výkresů. První a druhý sloupec je řešen pomocí checkboxů, jejichž zaškrtnutí v prvním sloupci znamená, že podle tohoto atributu mají být plochy rozpuštěny. Druhý sloupec určuje, zda má být atribut povinný. Význam obou checkboxů se pokusím vysvětlit na následující ukázce.



ukázka dvou objektů pro rozpuštění, přičemž první objekt obsahuje atribut COB a SKO, druhý pouze atribut SKO

<p>.....</p> <p>podle atrib. <input checked="" type="checkbox"/> COB <input type="checkbox"/> SKO</p>	objekty se rozpustí pouze v případě, pokud bude atribut COB prázdný
<p>.....</p> <p>podle atrib. <input checked="" type="checkbox"/> COB <input checked="" type="checkbox"/> SKO</p>	objekty se nerozpustí - druhý objekt nemá povinný atribut COB
<p>.....</p> <p>podle atrib. <input type="checkbox"/> COB <input checked="" type="checkbox"/> SKO</p>	objekty se rozpustí, jen pokud bude atribut SKO v obou totožný
<p>.....</p> <p>podle atrib. <input type="checkbox"/> COB <input checked="" type="checkbox"/> SKO</p>	objekty se rozpustí, jen pokud bude atribut SKO v obou totožný
<p>.....</p> <p>podle atrib. <input checked="" type="checkbox"/> COB <input type="checkbox"/> SKO</p>	objekty se rozpustí, pokud bude atribut COB prázdný
<p>.....</p> <p>podle atrib. <input checked="" type="checkbox"/> COB <input checked="" type="checkbox"/> SKO</p>	objekty se nerozpustí

Z ukázek vyplývá, že pokud je uveden atribut jako povinný a v jednom z objektů není, k rozpuštění nedojde. Pokud je zaškrtnuté pouze podle kterého atributu rozpustit a v jednom z objektů je a v druhém ne, záleží zda je atribut prázdný. Pokud je prázdný, rozpuštění se provede, v opačném případě k rozpuštění nedojde. V případě zaškrtnutí více checkboxů musí být platné všechny podmínky. Povinný atribut tedy funguje na principu selekce, protože po jeho zaškrtnutí jsou řešeny jenom ty plochy, který daný atribut obsahují.

Výběr a odznačení atributů, podle kterých rozpustit lze provést hromadně tažením myši nad jejich názvy. Dále tabulka obsahuje místní nabídku, která je dostupná po kliku pravého tlačítka myši nad oblastí tabulky. Nabídka obsahuje možnost vybrat *všechny* atributy, nevybrat *žádný* atribut a nebo provést *opak* výběru. Při těchto volbách je zaškrtnut vždy první checkbox pro výběr atributu, druhý checkbox určující zda atribut má být povinný lze ovládat pouze ručně. Pokud však není vybrán první checkbox a je zaškrtnut druhý, první se automaticky označí. Naopak odznačení druhého checkboxu je automaticky provedeno, pokud je odznačen checkbox první.

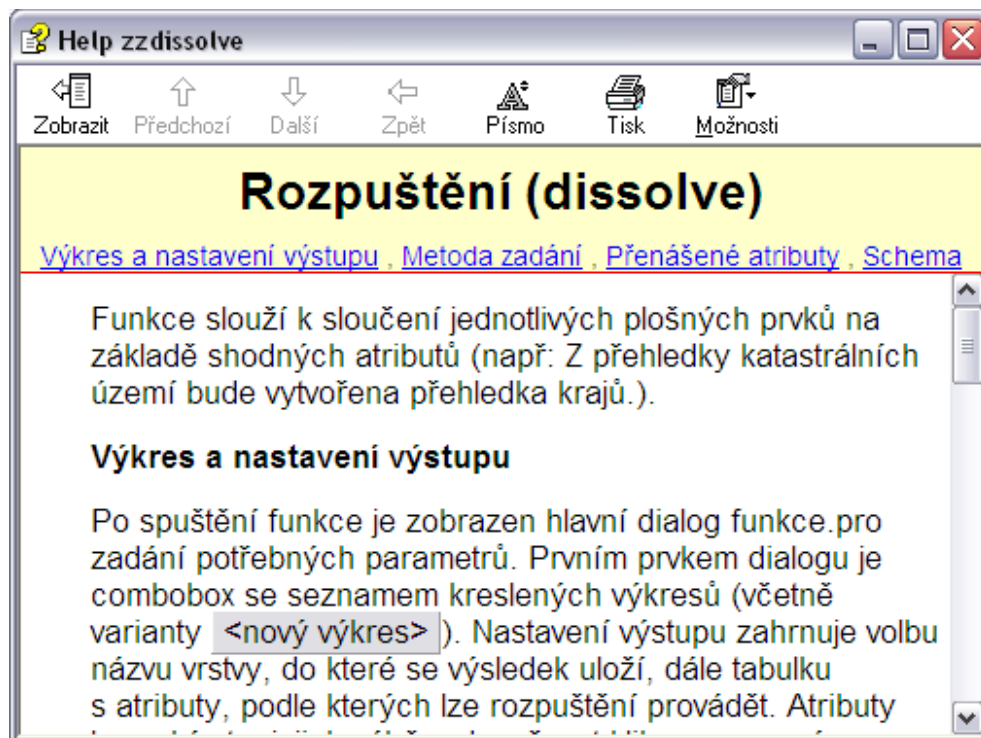
Další volbou je nastavení, zda plochám, které se nerozpouští (ale byly vybrány ke zpracování) *ponechat původní atributy*. Pokud nebude volba zaškrtnutá, atributy se nepřenesou, v opačném případě zůstanou plochám atributy původní.

Nastavení dialogu je ukládáno do inicializačního souboru do sekce [ZZDISSOLVE]. Princip je totožný jako u obalové zóny. Ukázka možného nastavení a jeho zápisu v souboru:

```
[ZZDISSOLVE] 'název sekce inicializačního souboru
NAST=34|17||25|10|PARSKUP;SKO|1;0| 'nastavení posledního stavu dialogu
VYK=C:\wkokes953\DEMO\LIP_STAN.VYK 'výkres pro výstup
NAST0=2|16||1|9|COB;SKO|0;0| 'hodnoty uloženého nastavení dialogu uživatelem
NAST1=2|18|COB|19|13|COB;CVK;CVO2;DRUPOZ_KN;SKO|0;1;1;1;0| '-||-
NAST2=34|1||25|10|PARSKUP;SKO|1;0| '-||-
saves=;nastaveni1;nastaveni2;nastaveni3 'seznam názvů uložených nastavení dialogu
```

K funkci byla napsána i nápověda (obr. 5.2), jejímž úkolem je seznámit uživatele s tím, co funkce dělá, co obsahuje a jak se ovládá. Obecnou nápovědu o funkci lze vyvolat pomocí tlačítka *Help*. Jednotlivé prvky dialogu, kromě typu text a frame, mají svůj tooltip, což je krátká nápověda popisující příslušný prvek. Tato nápo-

věda se zobrazí po najetí myši nad konkrétní prvek. Tooltips se zapisují přímo do zdrojového textu k definici dialogu, obecný help byl napsán v jiné aplikaci, jejímž výsledkem je nápověda ve formátu *chm*.



Obr. 5.2: Nápověda k funkci

Na stisku tlačítka *OK* je testováno, zda jsou zadané hodnoty korektní. Pokud jsou, funkce pokračuje rozpuštěním objektů, resp. výběrem oblastí či identifikací, v opačném případě je uživatel upozorněn následujícími hlášeními:

- Nebyla zvolena žádná cílová vrstva.
- Nebyl vybrán atribut pro rozpuštění ploch.
- Chybný kreslicí klíč.
- Nebyl vybrán obvod ani výplň plochy.

V současné době je ve společnosti GEPRO spol. s r.o. zpracovávána výkonná funkce pro rozpuštění, která převezme údaje zadané v popsaném uživatelském rozhraní. Po dokončení a otestování výkonné funkce bude Rozpuštění začleněno do standardní verze systému MISYS.

5.3 Možnost rozšíření o další funkce GIS

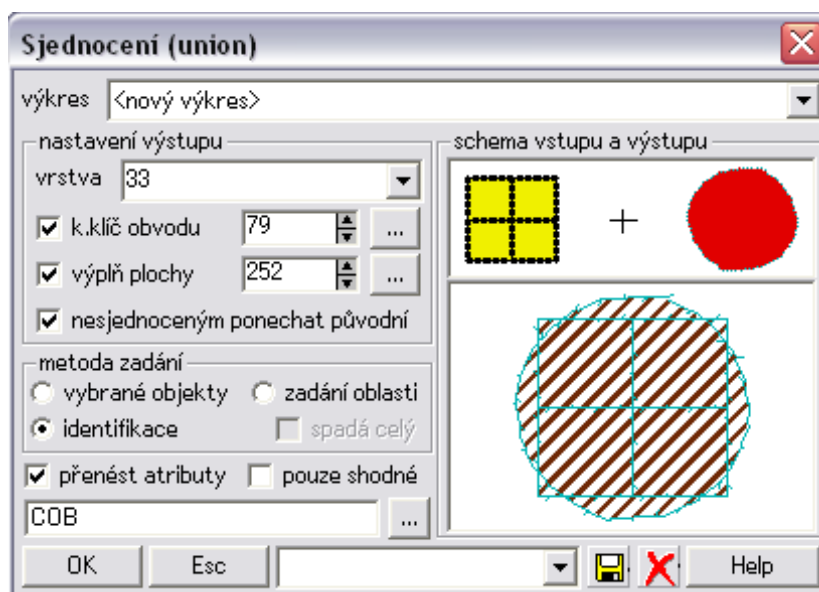
V této kapitole bych rád nastínil pár dalších funkcí patřících do kategorie překrytí, o které by měl být v budoucnu systém rozšířen a u kterých bude též potřeba vyřešit problematiku uživatelského rozhraní.

5.3.1 Sjednocení (Union)

Funkce slouží k sjednocení dvou nebo více plošných prvků, které se navzájem překrývají. Nejprve je spočítán průnik prvků, přičemž dojde k rozřezání polygonů. Poté jsou k nově vzniklým polygonům přiřazeny atributy (překrývající části mají atributy všech vrstev a nepřekrývajícím zůstávají atributy původní).

Příklad:

Vstupní data tvoří vrstva lesů a vrstva obcí ČR. Po provedení operace Union je vytvořena nová vrstva (vzájemně rozřezané polygony) obsahující atributy z obou vrstev. Díky tomu je následně možné např. zjistit, kolik procent z celkové výměry konkrétní obce tvoří lesní plochy. Hlavní rozdíl oproti operacím Intersect a Identifý je ten, že Union zachovává všechny prvky a všechny atributy obou vrstev [9].



Obr. 5.3: Hlavní dialog funkce Sjednocení

Po spuštění funkce je vyvolán dialog (obr. 5.3), který je velmi podobný (jak vzhledem tak ovládáním) dialogu z předcházející funkce. Rozdíl je v podstatě pouze v tom, že není potřeba pro tuto operaci volit atribut, podle kterého by se mělo rozpouštět. Dalším rozdílem je náhled v grafickém okně dialogu, který ukazuje dvě plochy a jejich sjednocení, přičemž je při kreslení přebírán aktuální kreslicí klíč obvodové linie a výplň plochy.

5.3.2 Ostatní funkce

Systém MISYS by mohl být v budoucnu doplněn i o další funkce z tabulky 5.1. Přístup k uživatelskému rozhraní by měl být podobný u všech těchto funkcí. Předpokladem je vyřešit zvlášť problematiku uživatelského rozhraní a zvlášť realizaci výkonné funkce. Narodil od funkce Rozpuštění a Sjednocení však nebude stačit zadat jednu datovou vrstvu, ale bude potřeba vyřešit způsob zadání dat pro dvě vrstvy. Důvod je ten, že je potřeba rozlišit, jaká vrstva bude vstupní (může obsahovat body, linie či plochy) a s jakou vrstvou se budou provádět zadané operace (vždy jen plošná vrstva).

K zadání dvou datových vrstev lze přistoupit několika způsoby:

Výkres - v dialogu by se zadávaly názvy dvou vstupních výkresů. Nezáleželo by na rozložení objektů do vrstev v rámci výkresu.

Téma - obsahuje libovolný počet výkresů. Zadání by bylo podobné jako v případě výkresu.

Vrstva - v tomto případě by byl zadáván pouze jeden vstupní výkres a názvy jeho dvou vrstev nebo dva výkresy a pro každý z nich jedna vrstva. Uspořádání dat ve výkresu by vyžadovalo striktní rozdělení do vrstev.

Selekce - vstupními daty by byla dvě grafická okna systému. Před spuštěním funkce by musela být provedena vhodná selekce v jednotlivých oknech.

Identifikace - jednalo by se o způsob totožný s řešením identifikace jedné ze zmiňovaných funkcí s tím rozdílem, že by musela být identifikace zadána dvakrát. Jednou pro vstupní data a podruhé pro data se kterými se bude zadaná operace provádět. Tento způsob není vhodný při větším počtu prvků.

Atribut - další možností by mohlo být zadání dvou atributů z jednoho výkresu nebo dvou atributů, každého z jiného výkresu.

Všechny uvedené způsoby vyžadují od uživatele vhodně připravená data. Vyjímkou by mohla být identifikace, která ale není příliš vhodná při větším počtu prvků. Pro zadání výkresů a grafických oken se selekcí by stačily dva comboboxy, u způsobu pomocí vrstev a atributů by musel být zadán jeden nebo dva výkresy, ze kterých se mají zvolené vrstvy nebo atributy uvažovat. Identifikace by mohla být řešena stejným způsobem jako u funkce Rozpuštění, Sjednocení či Obalové zóny s tím rozdílem, že by se muselo odlišit, pro jakou skupinu dat je právě identifikováno.

Možností by také bylo připravit uživateli více způsobů nebo je vhodně zkombinovat (např. zadání pomocí výkresu, vrstvy či atributu by šlo vzájemně propojit). Před předáním dat výkonné funkci by mělo být testováno, zda uživatel nezadal stejná data pro obě skupiny, stejně tak by výkonná funkce měla kontrolovat, zda nejsou prvky, u nichž se zvolená operace provádí, totožné².

K zadání dat lze přistoupit také podobným způsobem jako u předchozích funkcí. Zadání dat by mohlo být řešeno pouze jednou datovou vrstvou, tj. vybranými objekty či zadáním oblasti. Musela by být ale specifikována dodatečná volba (výkres, vrstva či jiné), která by určovala, jaká data z vybrané datové vrstvy budou považována za vrstvu, pro kterou provádět zadanou operaci. Tento přístup by měl zaručovat, že nedojde k výskytu duplicitních dat.

V této podkapitole jsem se snažil nastítnit způsoby, kterými by se mohlo zadání dat řešit. Tento závěr bude předložen odborníkům k vnitřní diskuzi. Není tedy vyloučeno, že bude navrženo ještě i jiné řešení.

²Netotožnost dat je vhodným předpokladem, ale není podmínkou

Závěr

Předkládaná diplomová práce se zabývá tvorbou uživatelského rozhraní pro nové GIS funkce. V případě atributové tabulky se jedná i o řešení výkonných funkcí, které jsou součástí samotného dialogu. Téma této práce vychází z požadavku uživatelů systému MISYS na rozšíření systému o tento typ funkcí. Cílem tedy bylo navrhnout rozhraní funkcí tak, aby bylo srozumitelné, obsahovalo potřebné zadání dat, zároveň bylo příjemné uživatelsky a vycházelo z jichž existujících postupů a zvyklostí systému MISYS.

U vytvořených funkcí je popisováno jejich ovládání a funkčnost, text je průběžně doplňován obrázky vzhledu uváděného řešení. V některých případech je uvedena část zdrojového kódu.

Hlavním přínosem diplomové práce jsou následující funkce:

1. Obalová zóna - řešeno kompletně uživatelské rozhraní, zadané údaje jsou následně předkládány výkonné funkci, která není součástí práce.
2. Atributová tabulka - funkce řešena v celém svém rozsahu, jelikož výkonné funkce jsou přímo součástí dialogu.
3. Rozpuštění (Dissolve) - návrh a řešení uživatelského rozhraní funkce.
4. Sjednocení (Union) - návrh a řešení uživatelského rozhraní funkce.
5. Návrh dalšího uživatelského rozhraní pro některé GIS funkce.

Všechny funkce byly naprogramovány v jazyce KOKEŠ Basic, který je součástí systému MISYS. Funkce pro tvorbu obalové zóny byla začleněna do systému a je od října 2009 součástí systému (verze MISYS 9.52). Dostupná je z položky menu **Výkres/Analytické funkce/Obalová zóna**. Atributová tabulka je ve fázi interního testování, pro Rozpuštění a Sjednocení jsou k hotovému uživatelskému rozhraní dodělávány výkonné funkce. Všechny tyto funkce by se po dokončení měly stát součástí standardní verze systému MISYS a měly by být začleněny i do geodetického programu KOKEŠ.

Je zřejmé, že vytvořené funkce budou srovnávány s GIS řešením od společnosti ESRI, které je v tomto oboru považováno za standard. Přístup k datům je však u obou systémů zcela odlišný, proto i uživatelské řešení je jiné.

Znalosti získané při realizaci uvedených funkcí a studiu řešené problematiky nepochybně využiji i pro svoji další odbornou činnost.

Použité zdroje

- [1] KOLÁŘ J., *Geografické informační systémy 10*, ČVUT Praha, 2003.
- [2] Environmental Systems Research Institute, Inc., *Understanding GIS*, USA, 1990.
- [3] RAPANT P., *Úvod do geografických informačních systémů*, VŠB – TU Ostrava, 2002.
- [4] HALOUNOVÁ L., *Přednáškové materiály GIS*, Přednášeno na ČVUT v Praze, letní semestr 2008.
- [5] ARCDATA PRAHA, *Komu je GIS určen?*, [online]. 2009, Dostupné z URL: <<http://www.arcdata.cz/uvod/>>.
- [6] GEPRO, *Geografické informační systémy, Geodézie a projektování*, [online]. 2007, Dostupné z URL: <<http://www.gepro.cz/>>.
- [7] ZIMMERMANN Z., *Programování tiskových funkcí v systému KOKEŠ Basic (bakalářská práce)*, ČVUT Praha, 2008.
- [8] JEDLIČKA K., *Přednáškové materiály - Úvod do GIS*, [online]. Dostupné z URL: <<http://gis.zcu.cz/studium/ugi/elearning/index1.htm>>.
- [9] BURIAN J., *Uříznout, odstranit, připojit, rozdělit, nahradit...* GeoBusiness 8+9, 2008.

Seznam symbolů, zkratek a pojmů

ČVUT	České vysoké učení technické
FSv	Fakulta stavební
GIS	geografický informační systém
MISYS	název programu - městský informační systém
KOKEŠ	systém pro geodézii a projektování
PROLAND	systém pro zpracování pozemkových úprav
SPI KN	soubor popisných informací katastru nemovitostí
KATASTR	aplikace pro práci s SPI KN
PASPORTY	aplikace umožňující připojit do systému MISYS řadu aplikací z různých oborů
ESRI	Environmental Systems Research Institute, přední mezinárodní společnost produkující software v oblasti geoinformačních technologií
ÚAP	Územně analytické podklady

Seznam příloh

A Příloha - ukázka zdrojového kódu

65

A Příloha - ukázka zdrojového kódu

Kód tvorby uživatelského rozhraní pro funkci Rozpuštění (Dissolve):

Pro přeložení makra je zapotřebí mít nainstalovanou podporu pro vývoj vlastních úloh v KOKEŠ BASIC (jedna z voleb při instalaci systému) kvůli vkládaným souborům, dále některé funkce dialogu vyžadují verzi systému MISYS vyšší než verzi 9.52 (kvůli souborům face.inc a face.bas). Jelikož některé části kódu se celé nevejdou na jeden řádek, je odřádkování naznačeno symbolem "->", jak na konci, tak začátku řádku.

```
#include <library\face.inc> 'dlg_error
#include <library\JJSSLB.INC> 'new_file
#include <windows.h> ' _SetFocus
#include <library\vtident.inc> 'id_input
#include <library\vtpolygn.inc> 'polygon_input_obj
#include <table.h>

const string T_novy_vyk_ import KOKMSG '<novy vykres>'
const string JJM627 import KOKMSG 'identifikace KK - z honza.bas
const string JJM628 import KOKMSG 'výběr z nabídky - z honza.bas
const string JJE252 import KOKMSG 'nebyla zadána cílová vrstva - z honza.bas
const string ZZide_p = "identifikace výplně plochy"
const string Qlinie import KOKMSG 'kvůli dotazu na identifikaci KK a identifikaci pro vybrání linií po OK.Click
const string Tvypln_plochy import KOKMSG 'vyplň plochy - kvůli dotazu na identifikaci P
const string TME58 import KOKMSG 'chybný kreslicí klíč"
const string ZZne_k_p = "Nebyl vybrán obvod ani výplň plochy." import KOKMSG
const string ZZno_ident = "žádný prvek nebyl identifikován"
const string ZZ_no_atr = "otevřené výkresy neobsahují žádné atributy pro spojení ploch"
const string Emalo_RAM import KOKMSG
const string TNastaveni import KOKMSG 'nastavení
const string TMM360 import KOKMSG " " již existuje."- z tomas.bas
const string TMM361 import KOKMSG "Přepsat ?"- z tomas.bas
const string ZZ_err_atr = "Nebyl vybrán atribut."
const string ZZ_err_atr1 = "pro spojení ploch."

#define DISS_ORIG 1 'ponechat puvodni
#define DISS_COPY_ATR 2 'prenest atributy
#define DISS_FENCE 4 'zadani oblasti
#define DISS_WHOLE 8 'spada cely
#define DISS_IDENTIFY 16 'identifikace
#define DISS_OUT 32 'vne oblasti
#define DISS_IDENTICAL 64 'pouze shodne

record dissolve_rec
  container cnr 'cilovy vykres
  string vrstva 'cilova vrstva
  string dynamic disatr 'pole s nazvy atributu podle kterych se spojuje
  int idisatr 'delka pole
  int dynamic atrflag 'atribut nutne vyzadovan? (1=vyzadovan,0=nevyzadovan), delka pole stejna jako u disatr
  string copy_atr 'prenasene atributy
  int flag 'ponecha symbologí,jestli prenaset atributy,zpusob zadani
  int klic '(=<0 znaci nekreslit obvod)
  int plocha '(<0 znaci nekreslit plochu)
  point dynamic pnts[0]
  int ipnts
  identify dynamic ide[0]
  int iide
end record
```

```

#define MF_STRING 0x00000000
#define MN_ALL 1 `vse
#define MN_REVERSE 2 `opak
#define MN_CLEAR 3 `clear

record rbutton `pro tvorbu PopupMenu nad tabulkou
  int Col `sloupec tabulky
  int Row `radek tabulky
  short xPos `x-souradnice kliku
  short yPos `y-souradnice kliku
  short RButtonType `typ RBUTTONTYPE_DOWN, RBUTTONTYPE_UP, RBUTTONTYPE_DBLCLK
end record

picture psave
  "eJzdkj0SgjAQhXekSCEzzljZqBfwaBbpgk6s8DAU6W08igewsJPCGdzd8LMrDTrQ+JjkDfsBG5K3"
  "gDVeE+hC0xLAOBbdBCoFwSxCa8w0jBWZc8y4ohiv7IM1ixUstFOoUU0/bEjlfVYeitGLZn7WqmdD"
  "uW6vftTNegGNPY4bVMkOUrhuqmR7KqUDO6Dnh+j1XXt+jJ6VI5w6pqJ/9C+yGH/dikqXPHMm81Zk"
  "ORd0oIVgfabGMzdzgjfjWZFFzxXF4ho1o8c1o296YgXtSNMvZtFhWHkq1mVjXtbudc+Gyfu7LL4B"
  "WMByyw=="
end picture

picture pdelete
  "eJyVks2K1EAUhW/rogkIBa7cSPsI7pWZV8pqZLBNlrhwaE+aGgbmEWbpQphBeuFDiGTEwzEILSQ"
  "xFtqeOsnVUnaJWnIPdxzcMr/T2gx/wTW7JXllPrVLYRS9ehnlTMrZBErVOSBOXeZG/r0uRTLASF"
  "B8mPkkNKckglj+OZDMoJNRlq09bbOpGGCilP5aCCFx9suTXxeCjNPato6omQdp4bJsae/zYkg2Iv"
  "bJPJEM/klZSvaPhiPs3L584Tw1RFS/8pN/CzeBOEd8NPzeRGuAWFbcN7QoXIWLaDJ6DcEbxp2wT4"
  "djqe7fjb5Iqd9ybHBX7TO62KF4Wmt21z/vpW027fvDzj+um7ftKXmu4+64WpuAqzMLUmxRWFrxho"
  "a7PCApvxvBAg3XJeGjrEmtTDPVvU1/SL9fLX/Qd/eNOtVy/PvmvOnZk071RQ9HtEOgD7WnsVI3oDL"
  "iMX4Xw+4jFhMkEQv8qaW81RiUR0MVf8YqoIav2aliYNOoiuilHg7VCMWx96cRUoH8mrSGW0zZTGR"
  "52iZsBjJ82rW8XCJkRIYjOQ5b8piJO//WLzGCf4wix9QFz0z9RFd0TFTX9Dd9ly/oi8tiz/RV5bF"
  "qjCwLNYwsAzWfLe1A84si13FBrOIGpVIEae4L5lFHGEpZhfXnQazWN53xrJYzlj8CzppYjE="
end picture

begin dialog atr_listbox_dis 0 0 80 98 "Přenést atributy"size
  checkbox cxshod 2 1 76 12 "&pouze shodné" #9 "Přenesou pouze shodné atributy z aktuálně vybraných atributů."
  listbox lb 2 13 76 69 "" #9 "Obsahuje seznam atributů kreslených výkresů." multi cxfloat cyfloat exact
  button ok 2 84 24 12 "&OK" #9 "Ukončí dialog a vybrané atributy vloží do editačního pole hlavního dialogu." yfloat
  button Esc 28 84 24 12 "&Esc" #9 "Ukončí dialog bez akceptování změn." 2 yfloat
  button Help 54 84 24 12 "&Help" #9 "Zobrazí obecnou nápovědu o funkci." 9 yfloat
sub atr_listbox()
  self.MinWidth=self.Width: self.MinHeight=self.Height
end sub
sub ok.Click()
  "" "ok"
  if (self.lb.Index < 1) && self.cxshod.Value==0 then
    dlg_error(ZZ_err_atr,self.Caption,self.Window)
  else
    DialogDestroy(self,1)
  end if
end sub
sub Help.Click()
  "" "HELP"
  ShowHelp()
end sub
end dialog

begin dialog ident_dlg_uni 0 20 58 30 "Identifikace" "" "dialog pro identifikaci"
  text txplo 4 4 18 12 "plocha"
  edit plo 26 3 30 12 "" #9 "Zobrazuje počet identifikovaných ploch." ronly
  button OK 2 17 26 12 "&OK" #9 "Ukončí dialog a rozpustí objekty podle vybraných ploch."
  button Esc 30 17 26 12 "&Esc" #9 "Ukončí dialog bez akceptování identifikace a vrátí se do hlavního dialogu."
sub Create()
  self.plo.Value="0"
end sub
sub OK.Click()
  "" "OK"

```



```

editint edp 64 138 40 12 ""#9"Kód výplně plochy rozpuštěných objektů. Lze použít i hodnotu 0 s významem ->
-> 'vyplnit barvou pozadí'.cxfloat
button btp 106 138 14 12 "...#9"Zadání hodnoty výplně plochy výběrem z nabídky nebo identifikací v grafice. ->
-> Výběr z nabídky není nabízen při volbě <nový výkres>."xfloat
checkbox cxpuv 6 151 110 12 "nerozpuštěný&m ponechat původní"#9"Určuje, zda nerozpuštěným objektům ->
-> ponechat původní atributy."cxfloat
frame fr2 2 169 122 27 "metoda zadání"cxfloat
radiobox rxall 5 174 62 10 "vy&brané objekty"#9"Rozpuštění pro všechny objekty aktuálního grafického okna."->
-> group
radiobox rxide 5 184 58 10 "i&dentifikace"#9"Rozpuštění identifikovaných ploch. Výběr ploch dostupný po stisku ->
-> tlačítka OK."
radiobox rxfence 68 174 54 10 "&zadání oblasti"#9"Rozpuštění ploch v zadané oblasti. Výběr oblasti dostupný ->
-> po stisku tlačítka OK."
checkbox cxcely 74 184 42 10 "spadá &celý"#9"Určuje, zda objekty částečně zasahující do vybrané oblasti mají být ->
-> zahrnuty do rozpuštění."
frame fr3 125 20 116 176 "schema vstupu a výstupu"xfloat
winbox win_vstup 128 24 110 85 ""#9"Schema dvou vstupních ploch."xfloat
winbox win_nov 128 109 110 85 ""#9"Schema výsledné plochy respektující kreslicí klíč a výplň plochy."xfloat
checkbox cxatrc 3 197 55 12 "přenést &atrib."#9"Určuje, zda přenést zadané atributy."
edit edatr 58 197 113 12 ""#9"Seznam přenášených atributů oddělený středníky."cxfloat
button btatr 173 197 12 12 "...#9"Zobrazí dialog na výběr atributů pro přenesení. V dialogu lze vybrat více ->
-> atributů."xfloat
checkbox cxshd 187 197 54 12 "pouze shod&né"#9"Přenesou pouze shodné atributy, které jsou uvedeny v editačním ->
-> poli."xfloat
button ok 2 210 38 12 "&OK"#9"Zkontroluje zadané hodnoty, ukončí dialog a pokračuje buď založením nového ->
-> výkresu, zadáním oblasti nebo identifikací a rozpuštěním objektů."
button Esc 42 210 38 12 "&Esc"#9"Ukončí dialog bez akceptování navolených změn v dialogu."
combobox cbsave 82 210 88 100 ""#9"Název uloženého nastavení dialogu. Obsahuje seznam dříve uložených ->
-> nastavení."dropdown cxfloat
checkbox cxsave 172 210 12 12 ""#9"Vyvolá dialog pro uložení nastavení hlavního dialogu."xfloat
checkbox cxsdel 186 210 12 12 ""#9"Smaže název aktuálně vybraného nastavení hlavního dialogu."xfloat
button Help 200 210 40 12 "&Help"#9"Zobrazí obecnou nápovědu o funkci."xfloat
container cbvyk_out 'aktualni vykres (kvuli televizi - $$THIS - a graf. nabidkam)
dissolve_rec r
string atrib 'atributy podane grp_attrib_names / potrebuju je jeste pro naplneni listboxu pro prenasene atributy
int tab_cell 'leva horni bunka, abych pri navratu do dialogu nastavil tabulku do takoveho stavu, ve kterem byla
int tab_row '-||-
int tab_col '-||-
sub dissolve_dlg() '""""""""""""""""konstruktor
  SetHelpContext(1)
  self.cxsave.Picture = psave
  self.cxsdel.Picture = pdelete
  self.MinWidth=self.Width
  self.MinHeight=self.Height: self.MaxHeight=self.MinWidth=self.Height
  self.edkk.Min = 0: self.edp.Min = -1
  self.tab.RowHeadWidth = 0
  self.tab.v.Width=self.tab.v.Width
end sub
sub kresli_vstup()
  object obj= self.win_vstup.Obj
  string info_puv1,info_puv2
  ObjOpenEdit(obj)
  HideObject(obj): clearObject(obj)
  info_puv1=";K7;P14"
  info_puv2=";K4;P4"
  'vrchni ctverecek
  ObjInsPoint(obj,-1,ukazka[0], 'E',info_puv1)
  ObjInsPoint(obj,-1,ukazka[1], 'L', "")
  ObjInsPoint(obj,-1,ukazka[4], 'L', "")
  ObjInsPoint(obj,-1,ukazka[5], 'L', "")
  ObjInsPoint(obj,-1,ukazka[0], 'L', "")
  'spodni ctverecek
  ObjInsPoint(obj,-1,ukazka[1], 'E',info_puv2)
  ObjInsPoint(obj,-1,ukazka[2], 'L', "")
  ObjInsPoint(obj,-1,ukazka[3], 'L', "")
  ObjInsPoint(obj,-1,ukazka[4], 'L', "")
  ObjInsPoint(obj,-1,ukazka[1], 'L', "")
  ShowObject(obj)
  rectangle rc = ObjRect(obj,1)
  rc.ExpandP(1.05): WinSetRect(self.win_vstup.Win,rc)

```

```

end sub
sub kresli_nov()
  int add
  string info_puv1,info_puv2,info_nov
  if !ObjOpenEdit(self.win_nov.Obj) then ObjOpenEdit(self.win_nov.Obj)
  HideObject(self.win_nov.Obj): clearObject(self.win_nov.Obj)
  if (self.cxkk.Value) then info_nov += ";K"+str$(self.edkk.Value) else add += 'L'
  if (self.cxp.Value) then info_nov += ";P"+str$(self.edp.Value)
  'novy obdelnik
  ObjInsPoint(self.win_nov.Obj,-1,ukazka[0],'E',info_nov)
  ObjInsPoint(self.win_nov.Obj,-1,ukazka[2],'L'+add,"")
  ObjInsPoint(self.win_nov.Obj,-1,ukazka[3],'L'+add,"")
  ObjInsPoint(self.win_nov.Obj,-1,ukazka[5],'L'+add,"")
  ObjInsPoint(self.win_nov.Obj,-1,ukazka[0],'L'+add,"")
  ShowObject(self.win_nov.Obj)
  rectangle rc = ObjRect(self.win_nov.Obj,1)
  rc.ExpandP(1.05): WinSetRect(self.win_nov.Win,rc)
end sub
sub recdlg() 'budu volat v Create a csave.Change (vykres,atribut,vrstva)
  int flag=FALSE,i,j
  'pouze stav posledniho ulozeni, nastaveni dialogu si nepamatuje nastaveni VYKu
  string pom=GetIniString(__KBE__,"VYK","")
  for i=0 to self.cbvyk.ListCount-1
    if self.cbvyk.List[i]==pom then self.cbvyk.Index=i: flag=TRUE
  next
  if flag=FALSE then self.cbvyk.Index=0
  self.elvrs.Out=self.r.vrstva 'vrstvu zde davam Outu az po naplneni In v Creatu
  if self.elvrs.Out="" then self.elvrs.Index=0 'pokud nebylo nic v ini, nastavim index na 0
  if pom="" then self.edkk.Value=1 'pokud nebyla funkce nikdy spustena,budu implicitne edkk nastaveno na 1
  int at
  'zaskrtnuti checkboxu v tabulce, v extract_ini sem se naplnil
  'radsi odskrtnu celou tabulku
  for i=1 to TabMaxRows(self,self.tab)
    TabGetRow(self,self.tab,i)
    'radsi si kazdy radek vynuluji a pak ho nastavim,pokud je hodnota v ini
    if self.tab.v.Value==1 then self.tab.v.Value=0: self.tab.vv.Value=0: TabSetRow(self,self.tab,i)
    for j=0 to self.r.idisatr-1
      if self.tab.l.Value==self.r.disatr[j] then
        if self.r.atrflag[j]==1 then self.tab.vv.Value=1
        self.tab.v.Value=1
      exit for
    end if
  next
  TabSetRow(self,self.tab,i)
next
end sub
sub extract_ini(string cis="")
  string ini,s
  int at,i,max
  ini=GetIniString(__KBE__,"NAST"+cis,"")
  self.r.vrstva = Strtok$(ini,',' ,at) 'vrstva (plnim az v Create)
  if (CheckInt(Strtok$(ini,',' ,at),i)) then i = 0 'vlajka
  self.r.flag = i
  if (i & DISS_ORIG) then self.cxpuv.Value = 1 else self.cxpuv.Value=0
  if (i & DISS_COPY_ATR) then
    self.cxatrc.Value = 1
    if (i & DISS_IDENTICAL) then self.cxshod.Value = 1
  else
    self.cxatrc.Value = 0
  end if
  i &= (DISS_FENCE|DISS_IDENTIFY)
  self.rxall.Value = 0: self.rxfence.Value = 0: self.rxide.Value = 0
  self.cxcely.Enable = 0: self.cxcely.Value = 0
  select case i
  case DISS_FENCE
    self.rxfence.Value = 1: self.cxcely.Enable = 1
    if (self.r.flag & DISS_WHOLE) then
      self.cxcely.Value = 1
    end if
  case DISS_IDENTIFY: self.rxide.Value = 1

```

```

case else: self.rxall.Value = 1
end select
self.edatr.Value = Strtok$(ini,'|',at) 'prenasene atributy
s = Strtok$(ini,'|',at) 'hodnota klice
if (asc(s)='#') then 'klic nezapnut
  self.edkk.Value = val(right$(s,len(s)-1)): self.cxkk.Value = 0
else
  self.edkk.Value = val(s): self.cxkk.Value = 1
end if
s = Strtok$(ini,'|',at) 'hodnota plochy
if (asc(s)='#') then 'klic nezapnut
  self.edp.Value = val(right$(s,len(s)-1)): self.cxp.Value = 0
else
  self.edp.Value = val(s): self.cxp.Value = 1
end if
'jdu si atributy podle kterych spojuji nahazet do pole
string pom_atrib = Strtok$(ini,'|',at) 'string z ini
i=0
int at1
do
  if (ubound(self.r.disatr,1)>=i) then max += 5: redim self.r.disatr[max]
  self.r.disatr[i]=Strtok$(pom_atrib,';',at1)
  if self.r.disatr[i]=" " then exit loop
  i+=1
loop
self.r.idisatr=i 'delka pole
'jdu precist ridici atributy
pom_atrib = Strtok$(ini,'|',at) 'string z ini
max=0: i=0: at1=0
do
  if (ubound(self.r.atrflag,1)>=i) then max += 5: redim self.r.atrflag[max]
  s=Strtok$(pom_atrib,';',at1)
  if s=" " then exit loop else self.r.atrflag[i]=val(s)
  i+=1
loop
end sub
sub save_ini(string cis="") '*****"save_ini
'melo by se volat v OK (bez parametru), v cxsave.Click a mozna i jinde
'napred aktualizuju vlaknu
self.r.flag &= !(DISS_COPY_ATR|DISS_FENCE|DISS_IDENTIFY|DISS_ORIG|DISS_IDENTICAL)
if (self.cxpuv.Value) then self.r.flag |= DISS_ORIG
if (self.cxatr.Value) then self.r.flag |= DISS_COPY_ATR
if (self.cxshod.Value) then self.r.flag |= DISS_IDENTICAL
if (self.rxfence.Value) then
  self.r.flag |= DISS_FENCE
  if (self.cxcely.Value) then self.r.flag |= DISS_WHOLE else self.r.flag &= !DISS_WHOLE
elseif (self.rxsode.Value) then
  self.r.flag |= DISS_IDENTIFY
end if
'vlastni zapis do ini:
'vrstva|flag|prenasene_atr|klic|plocha
'klic, plocha - pokud nema byt zapnuty, ulozit jako 0
string ini=self.elvrs.Out+"|"+str$(self.r.flag)+"|"+self.edatr.Value+"|"
if (self.edkk.Value >= 0) then 'mensi nez nula byt nemuze
  if (self.cxkk.Value == 0) then ini += "#"klic je vypnut (pise se jen pokud je zadan)
  ini += str$(self.edkk.Value) 'kreslici klic
end if
ini += "|"
if (self.edp.Value >= -1) then 'mensi nez -1 byt nemuze
  if (self.cxp.Value == 0) then ini += "#"plocha je vypnuta (pise se jen pokud je vypnuta)
  ini += str$(self.edp.Value) 'vypln plochy
end if
int i
ini+="|"
string pom,pom1
for i=1 to TabMaxRows(self,self.tab)
  TabGetRow(self,self.tab,i)
  if self.tab.v.Value==1 then
    pom+=self.tab.l.Value+"|"
    pom1+=str$(self.tab.vv.Value)+";"ukladam druhy sloupec

```



```

self.btkk.PopupMenu+=";"+JJM627 'identifikace KK
self.btp.PopupMenu+=";"+ZZide_p 'identifikace vyplne plochy
if (self.cbvyk.Index < (self.cbvyk.ListCount-1)) then
  self.cbvyk_out = GetContainer(self.cbvyk.Out)
else
  self.cbvyk_out = 0
end if
HideObject(self.win_nov.Obj)
if IsContainer(self.cbvyk_out) then
  self.btkk.PopupMenu+=";"+JJM628 'vyber z nabidky
  self.btp.PopupMenu+=";"+JJM628 'vyber z nabidky
  ObjSetAttrib(self.win_nov.Obj,"$$THIS",CnrGetName(self.cbvyk_out))
else
  ObjDelAttrib(self.win_nov.Obj,"$$THIS")
end if
ShowObject(self.win_nov.Obj)
self.cbsave.Out = "": self.cbsave.Index = -1 'zmizi nazev ulozeneho nastaveni
end sub
sub elvrs.Change() '*****ELVRS.CHANGE
  self.cbsave.Out = "": self.cbsave.Index = -1 'zmizi nazev ulozeneho nastaveni
end sub
sub fun_mn_all() '*****zaskrtne vsechny checkboxy v tabulce
  int i
  for i=1 to TabMaxRows(self,self.tab)
    TabGetRow(self,self.tab,i)
    if self.tab.v.Value!=1 then self.tab.v.Value=1: TabSetRow(self,self.tab,i)
  next
end sub //fun_mn_all
sub fun_mn_clear() '*****zaskrtne vsechny checkboxy v tabulce
  int i
  for i=1 to TabMaxRows(self,self.tab)
    TabGetRow(self,self.tab,i)
    if self.tab.v.Value!=0 then self.tab.v.Value=0: self.tab.vv.Value=0: TabSetRow(self,self.tab,i)
  next
end sub //fun_mn_clear
sub fun_mn_reverse() '*****provede opak stavu checkboxu v tabulce
  int i
  for i=1 to TabMaxRows(self,self.tab)
    TabGetRow(self,self.tab,i)
    if self.tab.v.Value!=1 then self.tab.v.Value=1 else self.tab.v.Value=0: self.tab.vv.Value=0
    TabSetRow(self,self.tab,i)
  next
end sub //fun_mn_reverse
sub intDlgProc(int hWnd, int Message, int wParam, int lParam) ' DLGPROC
  if ((Message == WM_COMMAND)&&(lParam == self.tab.Window)) then
    int wNotifyCode = wParam\65536
    if (wNotifyCode == SSN_BLOCKSELECTED) then
      'uzivatel oznacil pomoci klaves Shift a pohybu mysi (nebo
      'sipek nahoru/dolu) blok; nastane, az kdyz je Shift uvolnen.
      int i, r, c, value
      TabGetActiveCell(self,self.tab,r,c): i = r+1
      TabGetRow(self,self.tab,r): value = self.tab.v.Value
      while TabIsCellInSelection(self.tab.Window, c, i)
        TabGetRow(self,self.tab,i)
        if (self.tab.v.Value != value) then
          if self.tab.v.Value==1 then self.tab.v.Value=0: self.tab.vv.Value=0 else self.tab.v.Value=1
          TabSetDataCell(self.tab.Window,1,i,str$(self.tab.v.Value))
          TabSetDataCell(self.tab.Window,2,i,str$(self.tab.vv.Value))
        end if
        i += 1
      wend
      i = r-1
      while TabIsCellInSelection(self.tab.Window, c, i)
        TabGetRow(self,self.tab,i)
        if (self.tab.v.Value != value) then
          if self.tab.v.Value==1 then self.tab.v.Value=0: self.tab.vv.Value=0 else self.tab.v.Value=1
          TabSetDataCell(self.tab.Window,1,i,str$(self.tab.v.Value))
          TabSetDataCell(self.tab.Window,2,i,str$(self.tab.vv.Value))
        end if
        i -= 1
      end while
    end if
  end if
end sub

```



```

    end if
    self.cbsave.Out = "": self.cbsave.Index = -1 'zmizi nazev ulozeneho nastaveni
end sub
sub edatr.Change() '*****
    self.cbsave.Out = "": self.cbsave.Index = -1 'zmizi nazev ulozeneho nastaveni
end sub
sub btatr.Click() '*****BTATR
    atr_listbox_dis dlg
    dlg.Top = self.btatr.Top+self.btatr.Height: dlg.Left = self.btatr.Left
    dlg.lb.In = self.atrib
    if (self.cxatrc.Value) then dlg.lb.Out = self.edatr.Value else dlg.lb.Out = self.r.copy_atr
    repchar(dlg.lb.Out, ' ';': repchar(dlg.lb.Out, ' ';': repchar(dlg.lb.Out, '|';':')
    dlg.lb.Out = ";" +ucase$(trim$(dlg.lb.Out, ' ')) 'chci prave jeden strednik pred a zadny za
    'pozor: self.cxshod.Value muze byt zakazany, takze jeho .Value nemusí byt spravne
    if (self.r.flag & DISS_IDENTICAL) then dlg.cxshod.Value = 1 else dlg.cxshod.Value = 0
    if (DialogModal(dlg, 2, self) == 1) then 'OK
        self.edatr.Value = ltrim$(dlg.lb.Out, ' ')
        self.edatr.StartSel=0: self.edatr.EndSel=-1 'vybrany nazev automaticky posune v editu na konec textu
        self.cxatrc.Value = 1 'pokud neco vyberu, tak nebo zapnut, tak zapnu checkbox pro prenest atributy
        if (dlg.cxshod.Value) then self.r.flag |= DISS_IDENTICAL else self.r.flag &= !DISS_IDENTICAL
        self.cxshod.Enable = 1: self.cxshod.Value = dlg.cxshod.Value
        self.edatr.SetFocus()
    else
        self.btatr.SetFocus()
    end if
end sub
sub cbsave.Change() '*****CBSAVE.CHANGE
    int i = self.cbsave.Index
    if (i > -1) then
        self.extract_ini(str$(i))
        self.recdlg()
        self.cxsdel.Enable = 1
        self.cbsave.Index = i
        self.kresli_nov()
    end if
end sub
sub cxsave.Click() '*****CXSAVE
    int i
    string out=self.cbsave.Out
    i=dlg_savesetting(self.cbsave.In, self.btatr.Top, self.btatr.Left, out) 'fase.bas
    self.cxsave.Value = 0
    select case i
    case -1
        self.cbsave.Out=out: self.cbsave.In += ";" +out
        if(self.cxsdel.Enable!=1) then self.cxsdel.Enable = 1
    case >=0
        self.cbsave.Index=i
    case ESCAPE
        exit sub
    end select
    self.save_ini(str$(self.cbsave.Index))
end sub
sub cxsdel.Click() '*****CXSDDEL
    self.cxsdel.Value = 0
    if (self.cbsave.Index>-1) then
        int i = instr(self.cbsave.In+";";";"+self.cbsave.Out+";")
        if (i == -1) then exit sub 'nemelo by nastat
        'zrusim aktualni nabidku (RemoveItem nemodifikuje In, coz mi trosku vadi pri ukladani)
        string tmp$
        if (i > 0) then tmp$ = left$(self.cbsave.In, i)
        i += len(self.cbsave.Out)+2
        if (i < len(self.cbsave.In)) then tmp$ += mid$(self.cbsave.In, i, &hfff)
        'posunu udaje v INI
        for i = self.cbsave.Index to self.cbsave.ListCount-2
            PutIniString( __KBE__, "A"+str$(i), GetIniString( __KBE__, "A"+str$(i+1), ""))
        next
        PutIniString( __KBE__, "A"+str$(self.cbsave.ListCount-1), "") 'posledni smazu
        self.cbsave.In = tmp$
    end if
    self.cxsdel.Enable = (self.cbsave.ListCount>0)&1

```

```

    self.cbsave.Index = -1
end sub
sub cxshod.Click() '""""""""""CXSHOD
    if (self.cxshod.Value) then self.r.flag |= DISS_IDENTICAL else self.r.flag &= !DISS_IDENTICAL
    self.cbsave.Out = "": self.cbsave.Index = -1 'zmizi nazev ulozeneho nastaveni
end sub
sub ok.Click()
    string chyba = ""
    int i,focus=0, flag=FALSE,max,poc
    for i=1 to TabMaxRows(self,self.tab)
        TabGetRow(self,self.tab,i)
        if self.tab.v.Value==1 then
            if (ubound(self.r.disatr,1)>=poc) then max += 5: redim self.r.disatr[max]: redim self.r.atrflag[max]
            self.r.disatr[poc]=self.tab.l.Value
            if self.tab.vv.Value==1 then self.r.atrflag[poc]=1 else self.r.atrflag[poc]=0
            if flag!=TRUE then flag=TRUE
            poc+=1
        end if
    next
    self.r.idisatr=poc
    if flag==FALSE then chyba+=left$(ZZ_err_atr,len(ZZ_err_atr)-1)+ZZ_err_atr1
    'test, zda je zadana cilova vrstva
    if (self.elvrs.Out != "") then
        self.r.vrstva=self.elvrs.Out
    else
        if (chyba != "") then chyba += "#13"#10
        chyba+=ucase$(left$(JJE252,1))+mid$(JJE252,2,&hfff)+".(n->N)ebyla zvolena zadna cilova vrstva
        focus = self.elvrs.Window
    end if
    if (self.cxkk.Value|self.exp.Value) then
        if (self.cxkk.Value) then
            if (self.edkk.Value > 0) then
                self.r.klic = self.edkk.Value
            else
                if (chyba != "") then chyba += "#13"#10
                chyba += ucase$(left$(TME58,1))+mid$(TME58,2,&hfff) '(c->C)hybny kreslici klic
                if (focus == 0) then focus = self.edkk.Value
            end if
        else
            self.r.klic = 0
        end if
        if (self.exp.Value) then self.r.plocha = self.edp.Value else self.r.plocha = 0
    else
        if (chyba != "") then chyba += "#13"#10
        chyba += ZZne_k_p
        if (focus == 0) then focus = self.cxkk.Window
    end if
    'zajistim, aby string s prenasenymi atributy byl vzdy se stredniky
    self.r.copy_atr=repchar$(self.edatr.Value,',';)
    repchar(self.r.copy_atr,',';)
    repchar(self.r.copy_atr,',';)
    if (chyba == "") then
        self.save_ini() 'zapis do ini
        self.tab_cell=TabGetTopLeftCell(self.tab.Window,self.tab_col,self.tab_row)
        DialogDestroy(self,1)
    else
        dlg_error(chyba,self.Caption,self.Window)
        if (focus) then _SetFocus(focus)
    end if
end sub
sub Esc.Click() '""""""""""ESC
    DialogDestroy(self,2)
end sub
sub Help.Click() '""""""""""HELP
    ShowHelp()
end sub
end dialog

sub int dis_dlg(dissolve_rec& rdissolve)

```

```

identify dynamic id[0] 'vraci id_input
identify dynamic idee[10] 'si plnim sam kvuli objekty id kvuli rozsviceni a zhasnuti kresby
int i, j, max, pocet, k
dissolve_dlg d
ident_dlg_uni d_ident
id_set iv
object tmp_obj
string s=""
point p
group grpssel
window win = WinGetActive()
if IsWindow(win) then
  grpssel = WinGetGroup(win)
else
  grpssel = GrpCreate(): GrpInclAll(grpssel)
end if
'jdu kontrolvat, zda je nejaky atribut
if (grp_attrib_names(grpssel)== "") then print ZZ_no_atr: exit sub
d.cbsave.In = GetIniString( "__KBE__", "saves", "" ): d.cbsave.Index = -1
if (CheckInt(GetIniString( "__KBE__", "last_save", "" ), i) == 0) then
  if (i > -1) then
    d.cbsave.Index = i: s = str$(i)
  end if
end if
d.extract_ini()
do
chci_dlg:
  i = DialogModal(d,4) 'volam s flagem 4 kvuli nesknoceni dialogu po enteru nad jeho prvky
  d.r.vrstva=d.elvrs.Out
  select case i
  case 2 'Esc
    return -1
  case 1 'OK
    PutIniString( "__KBE__", "saves", d.cbsave.In): s = ""
    if (d.cbsave.Index > -1) then s = str$(d.cbsave.Index)
    PutIniString( "__KBE__", "last_save", s)
    rdissolve=d.r
    if (d.cbvyk.Out == T_novy_vyk_) then 'zalozeni noveho vykresu
      if new_file(rdissolve.cnr, "", NEWFILE_ONLYVYK OR NEWFILE_SKIPSAVE) then goto chci_dlg
    else
      rdissolve.cnr=GetContainer(d.cbvyk.Out)
      if !IsContainer(rdissolve.cnr) then return -1 'pro jistotu
    end if
    if (rdissolve.flag & DISS_FENCE) then 'omezeni oblasti
      InOutStat()
      if !IsObject(tmp_obj) then tmp_obj=ObjCreateTemp(1)
      i = polygon_input_obj(rdissolve.pnts,tmp_obj,vPLGi_o)
      if IsObject(tmp_obj) then DeleteObject(tmp_obj) 'zajisti mi, ze objekt nebude po funkci zustavat
      if (i > 0) then
        if TPolyOut then rdissolve.flag |= DISS_OUT 'vne
        if (i > 0) then rdissolve.ipnts=i: return 0
      end if
      'pri ESCAPE se to samo vrati k dialogu, pri SYSESCAPE to samo skonci
    elseif (rdissolve.flag & DISS_IDENTIFY) then 'identifikace
      int ident_co = IDENT_1|IDENT_AREA
new_ident:
      DialogModeless(d_ident)
      object obj_pom = ObjCreateTemp(1)
      ObjOpenEdit(obj_pom)
      'zajistim si, aby pri obnoveni okna mi to nespadlo kvuli neplatnemu popisovaci
      ObjClrFlags(obj_pom, OBJ_TEMP|OBJ_HELP)
      max=ubound(idee,1): pocet=0
      do
        i = id_input("", ident_co, id, iv)
        select case i
        case -1 'OK
          'test, zda je aspon neco identifikovano
          if (pocet == 0) then
            dlg_error(ZZno_ident, ""): goto new_ident
          end if

```

```

'kopirovani a pod.
rdissolve.iide=pocet
if (ubound(rdissolve.ide,1)<=pocet) then
  redim rdissolve.ide[rdissolve.iide]
  if (ubound(rdissolve.ide,1)<rdissolve.iide) then
    print _error(Emalo_RAM): j = -1: goto case_esc
  end if
end if
for j = 0 to pocet-1
  rdissolve.ide[j] = idee[j]
next
j = 0: goto case_esc
case <= ESCAPE
  j = -1
case_esc:
  DeleteObject(obj_pom)
  if DialogIsRunning(d_ident) then DialogDestroy(d_ident)
  if (i == ESCAPE) then exit loop
  return j
case else
  id[0].typ = ObjPointsLine(id[0].obj,id[0].bod)
  id[0].bod = ObjBeginLine(id[0].obj,id[0].bod)
  id[0].typ = -id[0].typ
  i = 0 'index bodu v pomocnem objektu
  HideObject(obj_pom)
  'prohledani jiz identifikovanych, zda tam dany prvek není
  for j = 0 to pocet
    if idee[j]==id[0] then 'prvek jiz byl identifikovan, tak ho vyhodim
      if id[0].typ<2 then d_ident.plo.Value=str$(val(d_ident.plo.Value)-1): idee[j].typ=-idee[j].typ
      'zrusim prislusne body v objektu
      for k = 0 to idee[j].typ-1
        ObjDelPoint(obj_pom,i)
      next
      WinRedraw(win) 'aby se pri ubrani prekreslila kresba
      ShowObject(obj_pom): pocet -= 1
      'posunu ostatni identifikace
      for j = j to pocet
        idee[j] = idee[j+1]
      next
      goto ven
    else
      i += abs(idee[j].typ)
    end if
  next
  if id[0].typ<2 then d_ident.plo.Value=str$(val(d_ident.plo.Value)+1): s = "L,3,3"
  ObjGetPoint(id[0].obj,id[0].bod,p)
  ObjInsPoint(obj_pom,-1,p,'E',s)
  for k = 1 to abs(id[0].typ)-1
    j = ObjGetPoint(id[0].obj,id[0].bod+k,p)
    if (j > 'Z') then j = 'L'
    ObjInsPoint(obj_pom,-1,p,j,"")
  next
  ShowObject(obj_pom): pocet += 1
  if (pocet > max) then
    max += 10: redim idee[max]
    if (ubound(idee,1)<max) then
      print _error(Emalo_RAM): j = -1: goto case_esc
    end if
  end if
  idee[pocet-1] = id[0]
end select
ven:
  loop
else
  return 0
end if
case 8 'zadani k.klice identifikaci
  iv.iv.ihelp = 2001: iv.iv.fhhelp = "ZLDISSOLVE"
  i = id_input(Qlimie,IDENT_LINE|IDENT_1,id,iv)
  if (i > 0) then 'neco identifikovano

```

```
    ObjPointValidK(id[0].obj,id[0].bod,i)
    if (i > 0) then d.edkk.Value = i 'nejaky klic se nasel
end if
case 9 'zadani plochy identifikaci
  iv.iv.ihelp = 2002: iv.iv.fhelp = "ZLDISSOLVE"
  i = id_input(get_question(Tvypln_plochy),IDENT_AREA|IDENT_1,id,iv)
  if (i > 0) then 'neco identifikovano
    s = ParseInfo('P',ObjGetPointInfo(id[0].obj,id[0].bod))
    if (s != "") then 'nejaka plocha se nasla
      i = clng(val(s))
      d.edp.Value = i
    end if
  end if
end select
loop while (i >= ESCAPE) 'pri i == SYSESCAPE koncim
return -1
end sub
```

```
macro dissolve help 1
  quickhelp "sloučení plošných prvků na základě shodného atributu"
  dissolve_rec rdissolve
  dis_dlg(rdissolve)
end macro
```