

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

EDITOR PASPORTIZACE VUT

SEMESTRÁLNÍ PROJEKT
TERM PROJECT

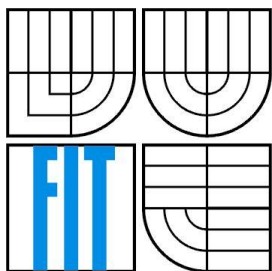
AUTOR PRÁCE
AUTHOR

Bc. Daniel Bierza

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

EDITOR PASPORTIZACE VUT
PASPORTIZATION EDITOR BUT

SEMESTRÁLNÍ PROJEKT
TERM PROJECT

AUTOR PRÁCE
AUTHOR

Bc. Daniel Bierza

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Jaromír Marušinec, Ph.D., MBA

BRNO 2007

Abstrakt

V této práci budu prezentovat problematiku pasportizace. Zaměřím se především na analýzu současného stavu v této problematice u budov VUT. Popíšu analýzu formátu „obr“ pomocí reverzního inženýrství. Navrhnou experimentální algoritmus pro převod formátu „obr“ na obecný značkovací jazyk a možnosti zobrazení grafických dat v závislosti na rychlosti a kvalitě vykreslení. Výsledkem bude převod formátu „obr“ do formátu XML a následné zobrazení.

Klíčová slova

Pasportizace, reverzní inženýrství, formát obr, možnosti zobrazení grafických dat.

Abstract

I will present the issue of passportization in my work. I will focus especially on the analysis of a current status of BUT buildings. The format "obr" analysis will be described by reverse engineering. I will design an experimental algorithm for conversion of the "obr" format into a universal markup language and then graphic data representation possibilities depending on speed and quality of its depiction. The data conversion from the "obr" format into the XML format and its projection will be the result of my work.

Keywords

Passportization, reverse engineering, format obr, graphic data representation possibilities.

Editor pasportizace VUT

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jaromír Marušinec, Ph.D., MBA.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Daniel Bierza
4. 1. 2008

Poděkování

Rád bych poděkoval vedoucímu mé diplomové práce za vypsání této práce, která je pro mne zábavou, a za inspirující připomínky. Objevuji nové věci, které budou pro můj další profesní růst přínosem. Děkuji i za pocit, že má diplomová práce nezůstane ležet v zásuvce, ale bude používána mnoha lidmi.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Pasportizace budov VUT	4
2.1	Apollo	4
2.2	GT facility	5
3	Reverzní inženýrství	7
3.1	Analýza formátu výkresů GT facility	8
4	Algoritmus převodu formátu obr na obecný značkovací jazyk	13
4.1	Postup při převodu formátu obr na XML	13
5	Návrh grafického prohlížeče	14
5.1	Volba grafické knihovny	14
5.2	Nastavení jakosti kreslení	15
6	Závěr	17
	Literatura	18
	Seznam příloh	19

Seznam obrázků

Obrázek 1 Grafická primitiva v Svopcad6	6
Obrázek 2 Základní operace s objekty.	6
Obrázek 3 Struktura souboru ve formátu obr.	9
Obrázek 4 Rozložení bitu určujících počet vrcholů.	10
Obrázek 5 Jednotná hlavička objektu uložených ve formátu obr.	11
Obrázek 6 Převod čtyř bytů obsahující souřadnici na číslo.	12
Obrázek 7 Diagram algoritmu pro převod z formátu obr na XML.	14

Seznam tabulek

Tabulka 1 Objekty uložitelné do souboru formátu obr.	8
Tabulka 2 Charakteristické znaky objektů.	10

1 Úvod

Tématem této práce bude problematika pasportizace budov. Pasportizace budov se zabývá shromažďováním technických, inženýrských, stavebních, architektonických, legislativních i provozních informací o jednotlivých objektech. V současné době vzrůstá význam pasportizace nejen pro společnosti spravující velké množství budov, pro developerské společnosti, ale i pro menší a střední organizace, které chtějí samy řídit provoz budov, plánovat rozvoj a mít objekty pod kontrolou. Při pasportizaci budov se mohou uchovávat nejen základní údaje o rozměrech a poloze budov a místností, ale i umístění rozvodů elektřiny, vody nebo plynu, údaje o vybavení místnosti, příslušnost elektrických obvodů k jističi a mnoho dalších údajů, které se vyplatí shromažďovat pro správu objektů.

Požadavkem na tuto práci je analyzovat software pro pasportizaci VUT. VUT používá pro pasportizaci program GT facility, který ukládá data do binárních souborů s koncovkou obr. Tyto data jsou provázána s databází, která uchovává popisné údaje o objektech. Důležitou součástí této práce bude analýza formátu obr souboru metodou reverzního inženýrství a zjištění vazeb na databázi. Výstupem této práce bude grafický prohlížeč, který umožní vstupní soubor ve formátu obr převést na obecný formát a v dalším kroku zobrazí získaná data.

V následující kapitole se budu věnovat popisu pasportizace budov VUT. Popíši programy svopcad6 a GT facility, které se používají jako editory pasportizace budov.

Třetí kapitola se věnuje reverznímu inženýrství. Pomocí reverzního inženýrství byla provedena analýza formátu souboru obr. Analýza formátu je stěžejní částí této kapitoly.

Kapitola následující po třetí kapitole se zabývá návrhem experimentálního algoritmu převodu formátu obr do formátu XML.

Poslední pátá kapitola popisuje návrh prohlížeče výkresů získaných převodem z formátu souboru obr.

2 Pasportizace budov VUT

Pro správu budov Vysokého učení technického se používá několik specializovaných programů. První z nich je program GT facility, který slouží jako grafický editor pasportizace budov a místností VUT. Program Apollo slouží jako prohlížeč pasportizace budov a místností VUT. Technologií databáze Oracle se má práce nebude zabývat. Napojení na databázi Oracle bude součástí diplomové práce. Nyní se zaměřím především na analýzu programu Apollo a GT facility, které jsou důležitým výchozím bodem pro analýzu souboru *.obr.

2.1 Apollo

Program Apollo obsahuje modul „Budovy a místnosti“, který je určen pro jednoduché prohlížení pasportizace budov a místností Vysokého učení technického. Obsahuje detailní přehled všech budov a místností včetně fotografií, výkresů objektů, leteckých snímků areálů a statistických informací. V programu se lze jednoduše přepínat mezi jednotlivými budovami Vysokého učení technického a zobrazovat jejich fotografie, letecké snímky a zobrazovat výkresy jednotlivých podlaží. Fotografie a letecké snímky nejsou dostupné ke všem budovám. Seznam objektů lze seskupovat a zobrazit podle různých kritérií. Základní seskupení je podle lokalit, kde je seznam rozdělen na Brno – Sever, Brno – Střed a Brno – Jih. Seskupení podle fakult zobrazí všechny fakulty VUT a seskupení podle areálů seskupí budovy podle ulic, na kterých sídlí.

Tento modul umožňuje i rychlé vyhledávání a zobrazení místností. Místnost lze jednoznačně vyhledávat podle jejího inventárního čísla, čísla místnosti či zkratky budovy. Další možností je vyhledávání místnosti podle osoby, která v ní sídlí nebo podle čísla telefonní klapky v místnosti. V tomto případě není úspěšné vyhledání zajištěno, protože mezi místnostmi a osobami/klapkami neexistuje jednoznačné propojení. Když v zobrazeném seznamu místností je nalezena požadovaná místnost, pak může být dvojklikem myši přímo zobrazena na výkrese k ní příslušející. Místnosti lze filtrovat podle jejich účelu. Lze zobrazit například jen hygienická zařízení, technické místnosti nebo kanceláře.

Ke každé místnosti zakreslené do výkresu jsou přiřazeny informace s ní spojené. Mezi základní informace, které lze získat o jednotlivých místnostech, patří číslo místnosti a číslo výkresu, ve kterém je místnost zakreslena. Údaj využívaný nejen při vyhledávání je název místnosti. U každé místnosti je zaznamenána plocha a výška. Mezi další významné údaje patří zařazení, účel místnosti a podlaží, ve kterém se místnost nachází. S každou místností může být

spojena i osoba, která má v místnosti pracovnu. Detailní informace o obsazení místnosti nejsou přiřazeny jednoznačně. Například v místnosti B206 pracuje ing. Petr Matoušek, Ph.D. Při získávání informací zjistíme nejen podrobnosti o jeho osobě, ale i o prodejně knih, sídlící v místnosti P206. Systém vyhledá pouze podle čísla místnosti. U každé místnosti lze editovat vybavení, které je v ní obsaženo. Vybavení místnosti lze vybrat pouze z pevného seznamu zařízení, mezi která patří bariera, bílá tabule, černá tabule, meotar, ozvučení, počítač, projektor, video a zatemnění. U vybavení lze nastavit, zda zařízení je přítomno nebo není, ale počet nebo typ zařízení již není možné nastavit.

Program Apollo umožňuje statistické vyhodnocení všech údajů, ke kterým přistupuje. Lze zjistit užitečnou plochu všech budov fakulty i počet obsazených místností fakulty. Tyto údaje lze zobrazit i v grafech.

Při zobrazení jakéhokoli výkresu se automaticky do složky Temp ukládá výkres ve formátu obr a z něj poté program získává informace a vykresluje je. Tyto soubory budou vstupem pro program převádějící binární soubor obr do čitelného formátu.

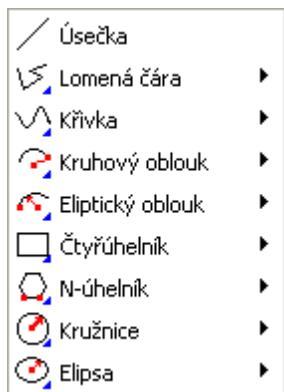
2.2 GT facility

Program GT facility slouží jako základní editor pro pasportizaci budov a místností Vysokého učení technického. V tomto editoru lze kreslit výkresy budov i dělat změny v již hotových výkresech. Tento program je propojen s databází a jakákoli změna se projeví okamžitě. Pro analýzu chování programu lze požit i program svopcad6, který je velmi podobný, ale nepracuje přímo s databází. Pro důležitost pasportizačních dat je vhodnější použít pro analýzu program svopcad6, který pracuje pouze se soubory a připojení k databázi lze simulovat lokální databází.

Primární určením programu svopcad6 je kreslení a editování dvourozměrných výkresů budov. Vstupem i výstupem editoru svopcad6 jsou binární soubory s koncovkou obr. Editor umožňuje vektorový exportovat do formátů dxf, vtx, emf, wmf, obx. Vektorový export ovšem nezachovává vazby s databází, uložené v souboru typu obr. Rastrový export umožňuje uložit výřez z právě zobrazené části výkresu do základních rastrových formátů. Důležitou součástí svopcad6 je import dat ze základních vektorových formátů mezi které patří dxf, dgn, shp, vtx, ocd, obx. Import se využívá pro vkládání nových výkresů dodaných k novým budovám.

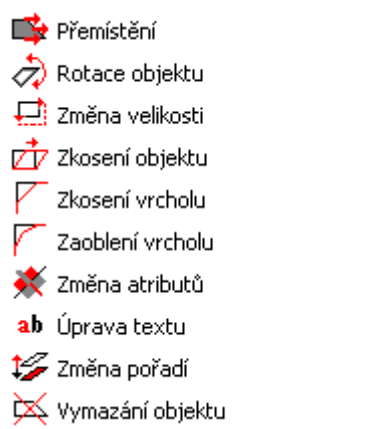
Svopcad6 umožňuje kreslení mnoha grafických primitiv (Obrázek 1), které mají mnoho dodatečných vlastností. Některé vlastnosti jsou u všech objektů stejné, například barva, tloušťka nebo typ vyplnění, který se zadává i u neuzavřených objektů. Důležitým parametrem u každého objektu je hladina, která udává příslušnost nakresleného objektu do skupiny. Hladinou

mohou být obvodové zdi, dveře, okna a další stavební prvky. Svopcad6 umožňuje zobrazit jen určité hladiny, což zrychluje například hledání chyb nebo úpravu jen určité skupiny objektů v budově.



Obrázek 1 Grafická primitiva v Svopcad6

Svopcad6 umožňuje základní operace s grafickými primitivami (Obrázek 2). Všechny operace však nejsou přístupné u všech objektů, například pro lomenou čáru nedokáže Svopcad6 provést operaci zkosení nebo zaoblení vrcholů. Pokud se objekt při rotaci ocitne částečně nebo zcela mimo povolené rozměry výkresu, tak zmizí a zobrazí se v poslední správné poloze až po použití lupy, která způsobí jeho znovu vykreslení.



Obrázek 2 Základní operace s objekty.

3 Reverzní inženýrství

Reverzní inženýrství je označován proces, jehož cílem je odkrýt princip fungování zkoumaného předmětu, jeho vlastností, vnitřních vztahů, architektury nebo designu. Reverzní inženýrství má za cíl sestavení ekvivalentního předmětu, který ale není kopií originálu. Mezi obvyklé postupy reverzního inženýrství patří měření a analýza struktury předlohy a zkoumání vnitřních vztahů částí systému. Výstupem reverzního inženýrství je obnovení části návrhu nebo jen dokumentace, které mohou být podkladem k vylepšení designu nebo například k bezpečnostnímu auditu. Reverzní inženýrství se nepoužívá jen v informatice, ale nachází uplatnění i v jiných lidských činnostech. V architektuře se reverzní inženýrství používá pro získání 3D modelu budov z naměřených významných bodů budov. Významné uplatnění našlo ve vojenství pro kopírování technologií. Sověti během let 1945 – 1949 „vyvinuli“ strategický bombardér Tupolev Tu - 4, který byl téměř dokonalou kopií amerického strategického bombardéru B-29. CIA se bránila proti úniku strategických technologií podstrčením několika technologií se skrytými chybami. V tomto případě se reverzní inženýrství stalo příčinou výbuchu sibiřského plynovodu, který po atomových výbuších je dosud největší umělá exploze vůbec.[3]

Reverzní inženýrství v informatice je definováno jako proces analýzy předmětného systému s cílem identifikovat komponenty systému a jejich vzájemné vazby nebo vytvořit reprezentaci systému v jiné formě nebo na vyšší úrovni abstrakce. V praxi jsou dva hlavní druhy. V prvním případě jsou dostupné zdrojové kódy, ale chybí nebo není kompletní dokumentace. Cílem reverzního inženýrství je v tomto případě vytvoření nebo zkvalitnění dokumentace k dostupným zdrojovým kódům. Tyto případy se musí často řešit i po odchodu klíčového zaměstnance. V druhém případě nejsou dostupné zdrojové kódy ani dokumentace. Tento proces se někdy nazývá „Reverse code engineering“. Reverzní inženýrství může být prováděno různými metodami. Jedna ze základních metod je pozorování skrze výměnu informací, která je nejčastěji používána v protokolovém reverzním inženýrství. Tato metoda sleduje například chování na sběrnici nebo v síti. Po provedení analýzy chování je možné vytvořit produkt napodobující chování vzoru. Druhou metodou je takzvaná „disassembly“, kdy se reverzní inženýrství provádí analýzou strojového kódu programu. Tato metoda je časově velmi náročná. Třetí často používanou metodou je „decompilation“. Tato metoda se snaží převést dostupný strojový kód programu do vyššího programovacího jazyku.

3.1 Analýza formátu výkresů GT facility

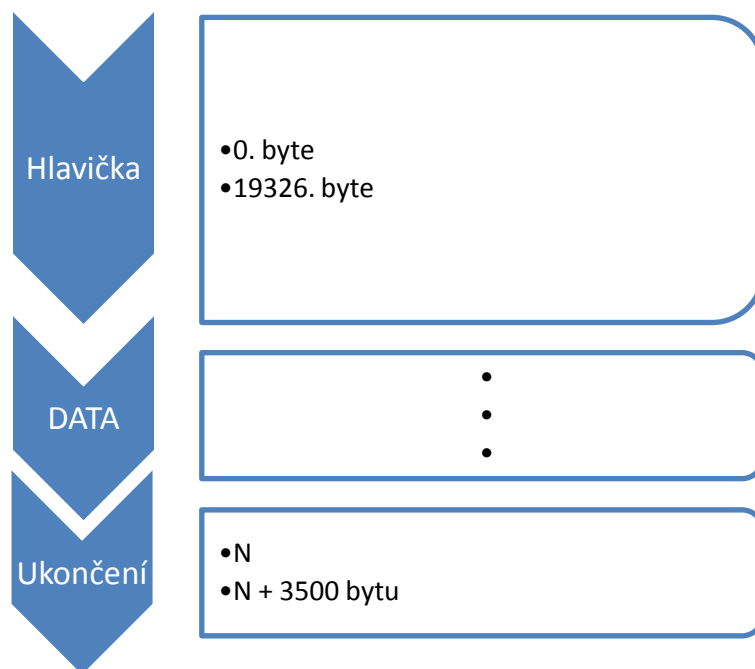
Pro analýzu formátu výkresů GT facility jsem použil reverzní inženýrství. Výkresy vytvořené v GT facility jsou uloženy do binárních souborů s koncovkou obr. Pro analýzu jsem ale použil program svopcad6, který ukládá do stejného formátu jako GT facility, ale není napojen na databázi a ukládá soubory pouze lokálně. Lokální uložení je výhodné ze dvou důvodů. Prvním důvodem snadnost přístupu k nakresleným výkresům, které se dají editovat ihned po uložení. Druhým důvodem jsou bezpečnostní omezení v GT facility. Při zásahu do existujících dat může dojít k nenávratné ztrátě velmi cenných dat. V programu svopcad6 lze vytvořit lokální databázi, tím se vytvoří stejné prostředí jako v programu GT facility.

Soubory ve formátu obr obsahují nejen popis nakreslených objektů, ale uchovávají i vazbu objektů na databázi. Prvním krokem v analýze bylo zjistit, jaké typy objektů lze do souboru uložit a jaké informace se u jednotlivých objektů uchovávají. Výkresy vytvořené v programu svopcad6 mohou obsahovat nejen jednoduché tvary, jako jsou čára nebo lomená čára, ale mohou obsahovat křivky, eliptické nebo kruhové oblouky, obdélníky nebo N úhelníky. Objekty, které lze nakreslit v svopcad6, se dají rozdělit na dva druhy. První typ objektu je s pevným počtem parametrů, které se ukládají do souboru (Tabulka 1). Druhý typ je hůře analyzovatelný, protože počet jeho parametrů může být proměnný (Tabulka 1). Proměnný počet parametrů způsobuje problémy při načítání, protože délka takového objektu uloženého v souboru může být téměř nekonečná. Svopcad6 nemá omezení v počtu parametrů ukládaných do souboru. Dobrým příkladem je křivka vytvořená tahem ruky. Při vytváření této křivky se vytváří při tahu myši lomená čára s velkým počtem vrcholů. Počet vytvářených vrcholů se při tahu postupně snižuje. Toto omezení není asi cílené, protože nižší počet vrcholů je způsoben pouze pomalostí programu svopcad6, který celou čáru překresluje při každém pohybu.

Objekty s pevným počtem parametrů	Objekty s proměnným počtem parametrů
čára	lomená čára
kruhový oblouk	pravoúhlá lomená čára
eliptický oblouk	křivka
obdélník	oblouková křivka
obdélník se zaoblenými hranami	křivka vytvořená tahem ruky
sít	křivka vytvořená analyticky
mnohoúhelník	
kruh	
elipsa	

Tabulka 1 Objekty uložitelné do souboru formátu obr.

Při analýze souboru jsem nejprve zjišťoval závislost typu uložených dat na pozici v souboru. Vytvořil jsem soubor ihned po spuštění programu svopcad6, ve kterém nebyly

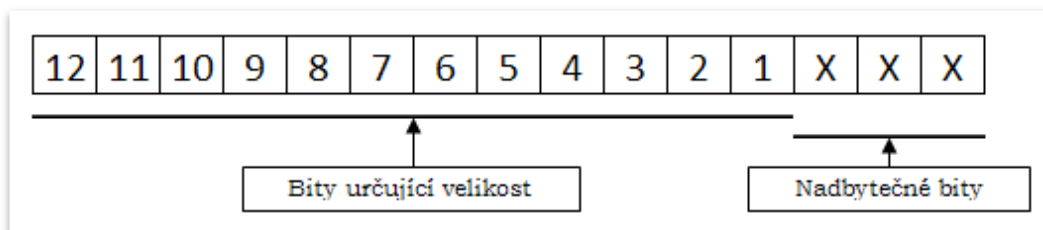


uloženy žádné objekty. Dalším krokem bylo postupně vytvořit všechny objekty, které svopcad6 umožňuje nakreslit. Každý soubor s takto vytvořeným objektem jsem porovnával se souborem neobsahující žádné objekty. Následující krok při zjišťování pozice ukládaných dat byl zjistit závislost množství uložených dat na struktuře souboru. Postup byl velmi podobný jako v předchozím případě. Porovnával jsem soubor neobsahující žádné objekty se soubory obsahující jeden až stovky objektů. Výsledky analýzy struktury souboru jsou pro načítání objektů velice příjemné. Soubor má neproměnnou strukturu (Obrázek 3). Začátek souboru až po pozici 19326 bytu od začátku je hlavička souboru. Na pozici 19327 bytu od začátku mohou začínat data uložených objektů. Ukončení souboru má vždy délku 3500 bytu.

Obrázek 3 Struktura souboru ve formátu obr.

Krok následující po zjištění struktury souboru byl zjištění rozdílů mezi uloženými rozdílnými objekty. Analýza rozdílů mezi jednotlivými objekty je velmi důležitá pro rozlišení objektu při načítání ze souboru uloženém ve formátu obr a následný převod. Každý objekt má specifický začátek, kterým začíná pole bytů, které obsahuje všechny informace o objektu. Tento začátek se skládá ze dvou bytů, které určují typ objektu. Všechny objekty, které jdou nakreslit v programu svopcad6, mají jiný specifický začátek (Tabulka 2), přestože mají některé úplně stejný obsah. Například oblouková křivka uchovává stejné parametry jako křivka, ale má jiný specifický začátek. Znakem každého objektu je i počet bytů ho popisující (Tabulka 2).

Objekty s pevným počtem parametrů (Tabulka 1) mají pevnou délku. Při načítání takového objektu lze načíst celý objekt najednou do paměti a poté převést jednotlivé uložené informace. Objekty, které mají proměnný počet parametrů (Tabulka 1), obsahují informaci o počtu uložených vrcholů. Číslo udávající počet vrcholů následuje po dvou bytech specifikujících typ objektu. Počet vrcholů je uložen na dvou bytech, ale první tři bity nenesou žádnou informaci (Obrázek 4). Maximální počet vrcholu je tedy 8191.

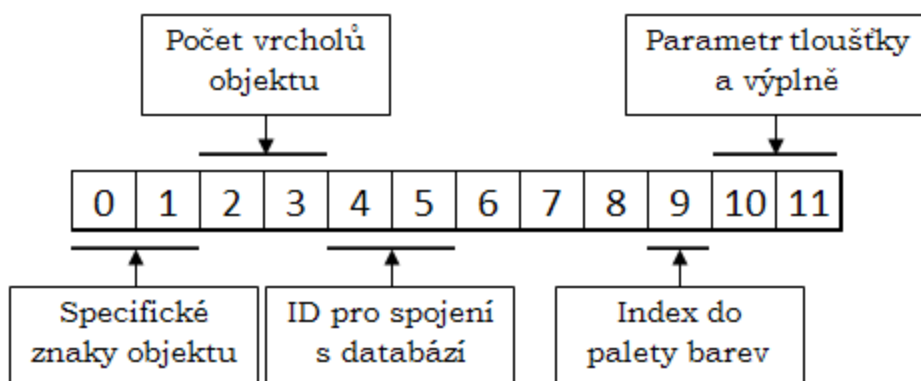


Obrázek 4 Rozložení bitu určujících počet vrcholů.

Objekt	Počáteční byty (hex)	Délka (byte)
bod	7017	28
čára	1815	28
lomená čára	8813	>= 46
oblouková křivka	8818	>= 52
pravoúhlá lomená čára	401F	>= 46
křivka	D417	>= 52
kruh	4416	36
elipsa	4516	36
kruhový oblouk	A816	44
eliptický oblouk	A916	44
obdélník	7C15	36
obdélník se zaoblenými hranami	8615	40
sít	7C17	40
mnohoúhelník	E015	38

Tabulka 2 Charakteristické znaky objektů.

U všech objektu v dalším kroku bylo třeba najít a převést všechny parametry. Některé parametry jsou u všech objektů stejné (Obrázek 5).

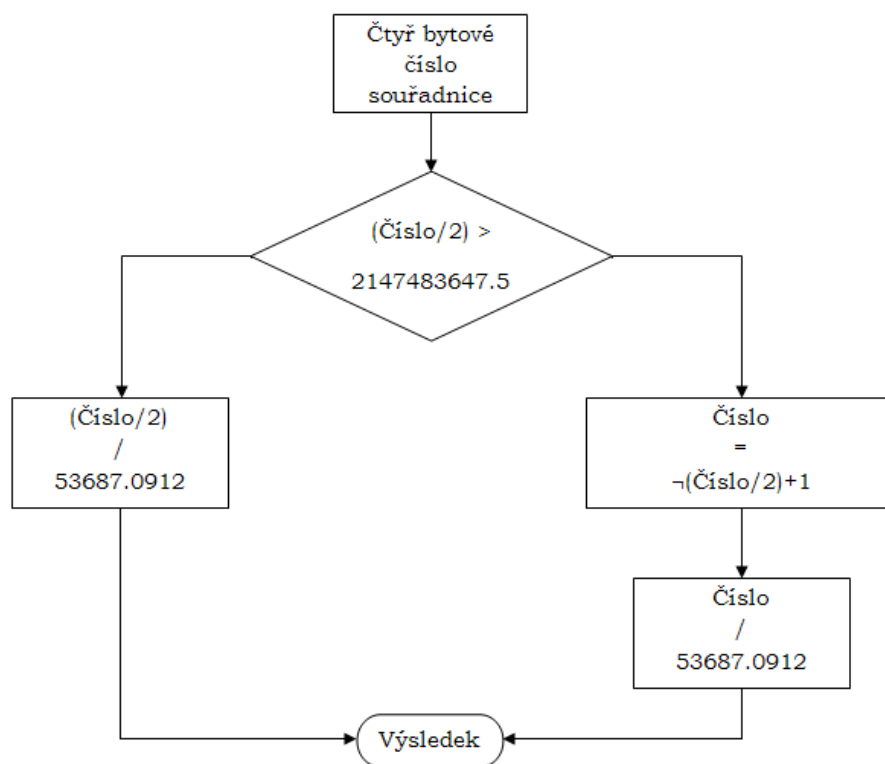


Obrázek 5 Jednotná hlavička objektu uložených ve formátu obr.

Specifické znaky u všech objektů jsou na prvních dvou bytech. Další dva jsou rezervovány pro počet vrcholů u objektů s proměnným počtem parametrů, například lomená čára nebo křivka. Na čtvrtém a pátém bytu je umístěno ID objektu pro vazbu s databází. Na devátém bytu je index do palety barev programu svopcad6. Tato paleta obsahuje 256 barev. Index do palety barev převádí na barvu uloženou pomocí tří složek RGB. Sice se ztrácí jeden z parametrů, ale pro pozdější zobrazení je tento převod nezbytný. Desátý a jedenáctý byte je rezervován pro tloušťku, styl čáry a výplň. Výplň se ukládá i u objektů jako je čára a další neuzavřené objekty. Informace o výplni je uložena na sedmém až jedenáctém bitu. Informace o tloušťce čáry je uložena na čtvrtém až šestém bitu a styl přerušování na nultém až druhém bitu. Třetí a dvanáctý až patnáctý bit nejsou využity. Přesto, že lze uložit zvlášť tloušťku a zvlášť styl čáry, tak editory svopcad6 a GT facility neumožňují nakreslit tlustší a zároveň čerchovanou čáru. Vždy jde vybrat jen jednu z možností.

Další vlastnosti nejsou u různých objektů vždy stejně rozmístěny a uloženy, ale u všech objektů se ukládá jeden nebo více bodů udávajících polohu. Každá bod je definován souřadnicí x a y . V editoru svopcad6 lze nastavit souřadnici nejmenší kladnou na hodnotu 0.001 a největší kladnou na 10 000. Stejný rozsah je i pro záporné hodnoty. Soubor uložený ve formátu obr má však souřadnici uloženou ve čtyřech bytech. Čtyři byty však odpovídají rozsahu 2 147 483.000 až -2,147,483.000. Soubor obr umožňuje uložit souřadnici v rozsahu 40 000.000 až -40 000.000 a ne rozsahu 10000, což povoluje program svopcad6, ani rozsah 2 147 483, což odpovídá čtyřem bytům, které se ukládají. Pro získání správné hodnoty je nutný převod, ve kterém se nachází několik na první pohled nejasných čísel (Obrázek 6). Protože je číslo uloženo v doplňkovém kódu, nejprve testují, v jaké části rozsahu se nachází. Pokud je číslo v rozsahu 0 až 2147483647.5, je to číslo kladné, ale pokud je číslo v rozsahu 2147483647.5 až 4294967295, je záporné a musí se převést na kladné. Záporné číslo se neguje a je k němu přičtena jednička (Pravá část diagramu). Po převodu na kladné číslo, pokud bylo záporné,

následuje vydělení kladného čísla číslem 53687.0912. Číslo 53687.0912 vzniklo vydělením čísla 2147483647.5 (maximální hodnota uloženého kladného čísla) číslem 40000 (maximálním číslem uloženým ve formátu obr). Výsledek se poté vynásobí -1, pokud bylo číslo záporné. Kdyby svopcad6 ukládal v rozsahu 10000.000 až -10000.000 bez nadbytečných převodů, vešlo by se číslo uchováající souřadnici do 7 bytů. Při použití 7 bytů by byla úspora jak místa, tak v rychlosti načítání. Jeden výkres části historického areálu kláštera fakulty informačních technologií obsahuje 2652 položek. Z převážné části jsou tyto položky čáry a lomené čáry s průměrným počtem bodu 6. Celkový počet uložených bodů je přibližně 8840. Každý bod se skládá ze dvou souřadnic, takže celková úspora místa je 17680B. Celkové velikost souboru je 137KB.



Obrázek 6 Převod čtyř bytů obsahující souřadnici na číslo.

Postup analýzy souborů ve formátu obr začínal zjištěním typů ukládaných dat, které program svopcad6 dokáže uložit. Následovalo nalezení pozice, kde se data v souboru ukládají. Nutným krokem pro správné načítání a rozpoznání objektů bylo rozpoznání rozdílů mezi uloženými objekty. Analýzou hlavičky pro všechny společné objekty jsem pokračoval. Následovala významná a časově náročná analýza uložených souřadnic bodů.

4 Algoritmus převodu formátu obr na obecný značkovací jazyk

Cílem experimentálního algoritmu pro převod souboru formátu obr je vytvořit soubor obecného značkovacího jazyka, který bude obsahovat všechny informace ze vstupního souboru. Převod nesmí ztratit žádnou informaci a nesmí žádnou informaci pozměnit. Rychlost algoritmu v tomto případě nepatří mezi nejdůležitější parametry pro hodnocení algoritmu, protože tento převod se bude dělat pouze jednou.

Vstupem do převodníku je soubor vytvořený v programu GT facility nebo svopcad6. Vstupní soubor obsahující výkres pasportizované budovy je uložen ve formátu obr. Výstupem algoritmu musí být soubor uložený v obecném značkovacím jazyku. Pro obecnost byl vybrán formát XML¹, který umožňuje snadné vytváření konkrétních značkovacích jazyků pro různé účely a široké spektrum různých typů dat.

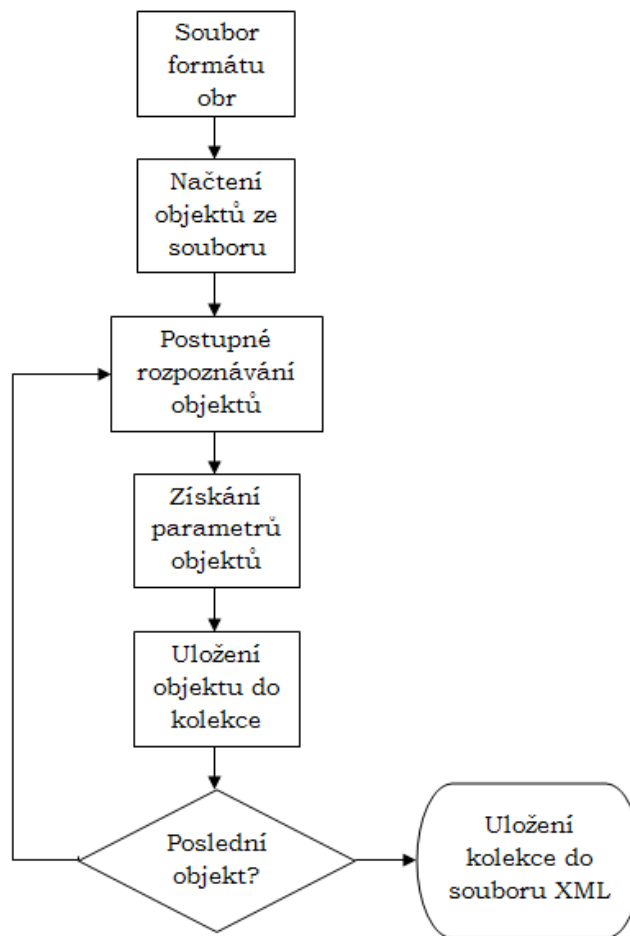
4.1 Postup při převodu formátu obr na XML

Převod souboru ve formátu obr na formát XML se skládá z několika kroků (Obrázek 7). Prvním krokem je načtení dat z binárního souboru ve formátu obr. Načítají se pouze data, ve kterých jsou uloženy objekty nakreslené ve výkrese. Počáteční hlavička souboru se nenačítá a ukončení taktéž. Pokud by se načítal celý program do paměti, zabírala by data, neobsahující žádné údaje, zbytečně velké množství paměti.

Dalším krokem je postupné rozpoznávání jednotlivých objektů. Každému objektu musí být přiřazen typ a podle typu se určí počet bytů, které objekt zabírá. Celý objekt se načte a jsou v něm postupně rozpoznávány všechny parametry, které příslušný typ objektu má obsahovat. Pokud je objekt kompletně rozpoznán, uloží se do generické kolekce, která se bude postupně plnit načítanými objekty.

Po načtení posledního objektu se celá kolekce uloží do souboru XML. K uložení používám XmlSerializer, který celou kolekci uloží ve formátu XML do zadaného souboru.

¹ eXtensible Markup Language



Obrázek 7 Diagram algoritmu pro převod z formátu obr na XML.

5 Návrh grafického prohlížeče

Hlavní důvodem pro vytvoření grafického prohlížeče je ověření správnosti převodu formátu obr do formátu XML. Pokud jednotlivé výkresy budov VUT obsahují tisíce objektů, je kontrola převedeného souboru bez grafického prohlížeče nemožná.

5.1 Volba grafické knihovny

Důležitým kritériem při volbě grafické knihovny pro vykreslování je typ vykreslovaných objektů. Pokud by grafický prohlížeč měl zobrazovat 3D data, byla by volba grafické knihovny zajisté jiná, ale pokud všechny objekty jsou pouze 2D, tak se výběr zaměří na jiné spektrum knihoven. Mezi knihovny, které jde použít pro vykreslování, patří Apple Quartz 2D, cairo nebo GDI+. Dva hlavní důvody pro výběr knihovny GDI+ jsou rozhodující a velmi zjednoduší

výběr. Prvním důvod je použití programu, který bude používán na počítačích se systémem Windows. Druhý důvod je, že celý program je implementován v jazyce C#.

Knihovna GDI+ vznikla pro systém Windows XP. GDI+ má vylepšené 2D grafické prostředí, funkce jako 2D Anti-aliasing, koordinaci pomocí plovoucí čárky, alpha blending, komplexnější správa. [5]

5.2 Nastavení jakosti kreslení

Nastavení jakosti kreslení má velký vliv na rychlost vykreslení grafických objektů a kvalitu zobrazení. Knihovna GDI+ má několik vlastností, které ovlivňují kvalitu vytvářené grafiky.

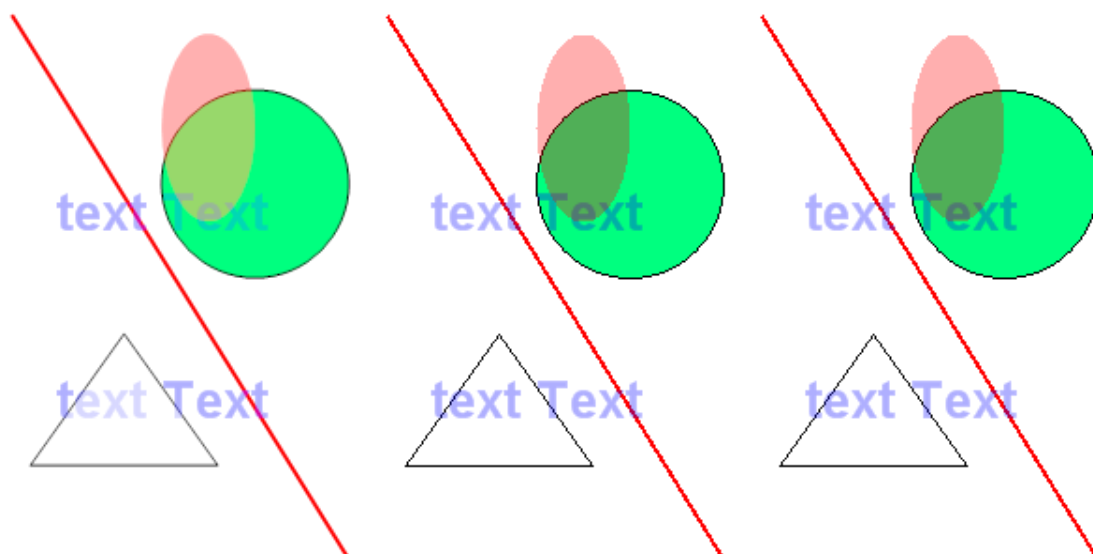
První takovouto vlastností je „SmoothingMode“, který nastavuje vyhlazování hran (antialiasing) při kreslení přímek a křivek. Při implicitním nastavení se hrany nevyhlazují, což způsobí, že šikmé přímky a křivky se nakreslí stupňovitě. Antialiasing způsobí, že místa napravo a nalevo vedle stupňů budou lehce stínovaná nebo vyplněna slabší transparentní barvou. Linie tak na pozorovatele působí hladším dojmem. Nejvyšší kvality dosáhneme při nastavení „HightQuality“ a největší rychlosti při nastavení „HightSpeed“. [4]

Dalším parametrem rozšiřujícím nastavení antialiasingu je „PixelOffsetMode“. Tento parametr nastavuje posunutí pixelu, čímž dosahuje změkčení stínů. Nejvyšší kvality dosáhneme při nastavení „HightQuality“ a největší rychlosti při nastavení „HightSpeed“ podobně jako u vlastnosti „SmoothingMode“. [4]

Vlastnost „InterpolationMode“ určuje interpolaci, která se používá při zvětšování nebo zmenšování obrázku. Nemá žádný vliv na kreslení grafických objektů jako přímek, obdélníků a kruhů. Nejlepší kvality dosáhneme pomocí parametru „HightQualityBicubic“. [4]

Vlastnost „CompositingQuality“ ovlivňuje kvalitu, pokud se kreslí transparentní grafické objekty. Při nastavení vysoké kvality parametrem „HightQuality“ je kontrast transparentně kreslených objektů podstatně vyšší než při nízké kvalitě. Nízká kvalita nastavená parametrem „HightSpeed“ vede k výraznému zrychlení. [4]

Srovnání vykreslených objektu jde vidět na obrázku (Obrázek 8). Defaultní nastavení kvality vykreslení grafických objektů má stejnou kvalitu jako nastavení HightSpeed. Nej kvalitnější je první z trojice obrázků, u kterého všechny parametry byly nastaveny na nejvyšší rychlost.



Obrázek 8 Různá kvalita vykreslení. HightQuality, HightSpeed, Default.

Rychlost vykreslených dat je závislá na kvalitě nastavení vykreslování. Srovnání dopadlo podobně jako v předchozím případě (Tabulka 3). Obrázek (Obrázek 8) byl pro každou kvalitu vykreslen 5000x. Výsledné časy jsou dosti rozdílné, pokud porovnáváme vysokou kvalitu s defaultním nastavením a vysokou rychlostí. Rychlost u Default a HightSpeed je však stejná.

Kvalita	Čas
HightQuality	44,98s
HightSpeed	17,91s
Default	16,31s

Tabulka 3 Srovnání časové závislosti na kvalitě.

6 Závěr

V semestrálním projektu jsem se zabýval pasportizací budov VUT. VUT používá program GT facility jako editor výkresů pro pasportizaci, který výkresy budov ukládá do binárního souboru formátu obr. Cílem mé práce bylo nejprve analyzovat program GT facility. Při analýze jsem zjišťoval všechny možnosti, které program poskytuje. Důležitým výsledkem byl seznam všech objektů, které lze zakreslit a parametry, které se k jednotlivým objektům ukládají. Tyto údaje byly důležitým krokem v dalším úkolu, kterým byla analýza formátu obr pomocí reverzního inženýrství. Při analýze souboru jsem postupně odhaloval umístění ukládaných dat, rozdíl mezi jednotlivými objekty, hlavičku objektů podle které se objekty dají rozřadit. Přicházel jsem na komplikovaný způsob uložení souřadnic. Pro propojení objektu s databází bylo nutné vypátrat umístění ID u všech typů objektů. Z analýzy formátu obr jsem mohl odvodit algoritmus, který data z jakéhokoli souboru formátu obr dokáže načíst. Výstupem algoritmu je kolekce objektů, která se může uložit do souboru formátu XML. Tato část práce je velmi významná, protože binární soubor, který nebyl pro nikoho čitelný, se po převodu do XML stává přehledným a srozumitelným. Převodem do XML se zbavujeme závislosti na programu GT facility, který je velmi nespolehlivý a špatně dokumentovaný. Při analýze GT facility bylo velmi obtížné vytvořit některé typy objektů, protože ovládání programu není příliš intuitivní. Po převedení a uložení souboru bylo nutné pro kontrolní účely navrhnout grafický prohlížeč převedených dat. Nejprve jsem hledal vhodnou grafickou knihovnu pro zobrazování 2D grafických dat. Pro mne byla nejvhodnější knihovna GDI+. Poté jsem provedl analýzu vztahu rychlosti vykreslovaných dat na kvalitě. Tento údaj bude potřebný pro vhodné vykreslení i výkresů, ve kterých je obsaženo mnoho objektů.

Práce, která mne ještě čeká, je návrh nového uspořádání parametrů u grafických objektů, protože aplikace GT facility některé parametry nevyužívá a některé neukládá. Navržení vhodného editoru pasportizace a vhodné uložení dat.

Literatura

- [1] Vysoké učení technické v brně - Apollo nápověda, Dokument dostupný na URL <http://is.vutbr.cz/napoveda/apollo.php> (4. 1. 2008)
- [2] Rivero C., Doorn H., Ferraggine E.: Encyclopedia database technologies and applications, 1, London, Idea Group Inc. 2006
- [3] Reverse engineering, Dokument dostupný na URL http://en.wikipedia.org/wiki/Reverse_engineering (4. 1. 2008)
- [4] Bayer J.: C# 2005 Velká kniha řešení, 1, Brno, Computer Press, a.s., 2007
- [5] Graphics Device Interface, Dokument dostupný na URL http://en.wikipedia.org/wiki/Graphics_Device_Interface (4. 1. 2008)

Seznam příloh

Příloha 1. CD obsahující text semestrálního projektu, zdrojové texty program a funkční program pro převod, uložení a zobrazení dat ze souboru formátu obr.