

1. Úvod a formulace problému

Tato diplomová práce je součástí rozsáhlejšího projektu nazvaného **Živé divadlo**. Projekt je veden Ing. Jindřichem Hodačem, Ph.D. z Katedry mapování a kartografie Fakulty stavební ČVUT za podpory Nadace barokního divadla a jeho cílem je vytvořit prostorový informační systém barokního divadla na zámku v Českém Krumlově, který by se dále využíval jak pro prezentaci divadla, tak pro jeho studium. Dalším cílem projektu je prozkoumat možnosti takových informačních systémů a sestavit do budoucna úspornou technologii jejich tvorby. Podrobnější informace o projektu viz [1].

1.1 Výchozí stav a cíl práce

Tato kapitola ve stručnosti shrnuje, do jaké fáze se práce na prostorovém modelu barokního divadla dosud dostala, a zároveň naznačuje hlavní problémy, které je v současném stavu třeba řešit (a jejichž řešení bude náplní této diplomové práce).

1.1.1 Motivace: stav prostorového modelu barokního divadla v Českém Krumlově

V létě roku 2004 bylo možno považovat měřické (terénní) i kancelářské práce na prostorovém modelu barokního divadla za ukončené. Prostorový model byl konstruován v CAD systému MicroStation ve formátu DGN.

Jedním z hlavních problémů vytvořeného modelu byla nekompatibilita jednotlivých jeho částí, které byly vytvářeny v rámci diplomových prací různých studentů a v různé době, aniž by byl brán ohled na celkovou soudržnost modelu. Jednotlivé části modelu jsou sice vytvořeny v témže souřadnicovém systému, nicméně dosud nebyla nijak ošetřena jejich návaznost. Kromě toho měl každý z tvůrců modelu svůj vlastní způsob práce v CAD systému, díky čemuž jsou podobné prvky modelu často konstruovány zcela různými způsoby. Hotový model tedy musel projít závěrečnou redakcí, která sjednotila formu všech částí modelu a vyřešila oblasti jejich styku.

Dalším krokem na cestě k prostorovému informačnímu systému je převod CAD modelu do

nějakého specializovaného formátu určeného pro popis virtuální reality. Samotný prostorový CAD výkres je totiž statický: umožňuje pouze prohlížení modelu, případně jeho řízenou animaci; abychom měli možnost pracovat s modelem interaktivně, tj. procházet jím, zobrazovat si doplňkové informace, manipulovat s předměty apod., potřebujeme jej exportovat z CAD formátu do formátu speciálně určeného pro popis takových modelů.

1.1.2 Cíle této diplomové práce

Cíle této práce již částečně vyplývají z předchozích odstavců. Jsou to:

- (a) doměření a konstrukce modelu několika zbývajících menších prostor divadla,
- (b) konečná redakce modelu, tj. sjednocení názvosloví souborů a vrstev a kontrola návaznosti jednotlivých částí modelu,
- (c) příprava celého modelu na převod do zvoleného formátu pro tvorbu interaktivních virtuálních modelů a
- (d) návrh konverzního programu pro výše uvedený převod.

Ad (a): předchozí diplomové práce byly věnovány těm prostorům divadla, které jsou významné právě pro funkci budovy jakožto divadla, nebo jsou podstatné konstrukčně – hlediště, jeviště, horní a dolní mašinerie, krov. **Ostatní vnitřní prostory divadla jsou prostorami spíše provozními, nemají tak velký význam a jejich měření a modelování v CAD systému bylo jedním z mých úkolů.** Totéž platí o exteriéru divadla (a exteriéru Renesančního domu, který na budovu divadla přímo navazuje).

Ad (b): protože byly jednotlivé části modelu vytvářeny různými lidmi, bylo třeba vtisknout DGN modelu jednotnou formu. Prvním krokem byla **reorganizace souborů, z nichž se model skládá:** prostory divadla (modelu) dostaly pořadová čísla a každý takto vymezený prostor je nyní obsahem jednoho souboru, jehož název začíná právě tímto pořadovým číslem. Druhým (velmi významným) krokem byla **reorganizace vrstev modelu.** Zavedl jsem jednotné názvosloví vrstev pro všechny části modelu a v některých případech i přeskupil jejich obsah. Konečné uspořádání CAD modelu je uvedeno v přílohách č. 1 (seznam DGN souborů) a č. 2 (seznam vrstev).

Poté bylo možno přikročit ke **kontrole a opravě oblastí, kde se části modelu stýkají.**

Ad (c): hotový model ve formátu DGN je nyní třeba **převést do vhodného formátu pro popis virtuální reality**. O výběru vhodného formátu pojednávám v kapitole 1.3.

Během práce ovšem vznikla otázka, zda je rozumné provádět konverzi tohoto konkrétního modelu, aniž bychom se technologii převodu dat pokusili nějak zobecnit a využít i pro jiné modely. Tak vzniklo zadání hlavního úkolu této diplomové práce: návrh konverzního programu, který by dokázal vytvořit ze statického prostorového CAD výkresu interaktivní virtuální model.

S tím souvisí další otázka, totiž zda je formát DGN, v němž byl model divadla vyhotoven, vhodným vstupním formátem této konverze. Jestliže je úkolem vytvořit poměrně složitou aplikaci, bylo by záhodno, aby její vstupní formát byl pokud možno univerzální, tj. aby soubor vstupující do konverze bylo možné vytvořit v různých CAD systémech.

Odpověď je zřejmá: nejpoužívanějším vektorovým formátem dat je jednoznačně formát DXF; drtivá většina CAD systémů s ním dokáže pracovat a je to navíc formát otevřený.

Nejrozsáhlejším úkolem při závěrečných úpravách modelu se tak stal export z DGN do formátu DXF. K této problematice se ještě dostaneme, nyní jen stručně: ne všechny prvky formátu DXF je možno použít pro konverzi; z toho důvodu je třeba dát DGN modelu takovou formu, aby se v něm tyto nepoužitelné prvky po exportu do DXF nevyskytovaly. Protože export z MicroStationu do DXF je vestavěnou funkcí programu, nedokážeme jej příliš ovlivnit a musíme tedy nejprve zjistit, jak přesně pracuje, tj. které grafické prvky v DGN převede na které prvky v DXF. Potom teprve můžeme model upravit tak, aby neobsahoval žádné “závadné” prvky. Více o této fázi práce viz kap. 2.5.

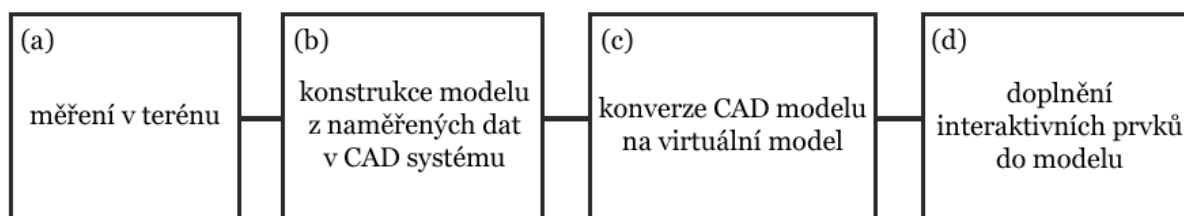
Ad (d): hlavní a nejprínosnější náplní této diplomové práce je pak **návrh konverzního programu z formátu DXF do zvoleného formátu pro virtuální realitu**. Návrhem programu rozumíme sestavení vývojových diagramů (s výjimkou detailních výpočtových algoritmů) a návrh uživatelského rozhraní (tj. soustavy dialogových oken, z nichž bude konvertor sestávat).

1.2 Technologie tvorby prostorových modelů

V této kapitole se nejprve pokusím popsat práci na virtuálním modelu budovy jednoduchým všeobecným schématem. Poté přikročím k nejdůležitějším požadavkům, které na výsledný virtuální model obecně klademe.

1.2.1 Obecné schéma tvorby virtuálního modelu

Práci na virtuálním modelu můžeme stručně popsat následujícím schématem:



Obr. 1.1: Obecné schéma tvorby interaktivního virtuálního modelu

Ad (a): zaměření objektu vhodnou metodou, ať už geodetickou, fotogrammetrickou nebo jinou. Popisy měřických prací lze nalézt např. v předchozích diplomových pracích uskutečněných v rámci projektu Živé divadlo (viz [3], [4]).

Ad (b): postup konstrukce modelu v CADu nemůže být zcela libovolný, protože pozdější konverze CAD modelu na virtuální model bude pravděpodobně vyžadovat, aby měl určitou předem stanovenou formu.

Ad (c): konverze musí brát ohled na požadavky, které klademe na virtuální modely obecně, viz kap. 1.2.2.

Ad (d): interaktivními prvky se rozumí např. možnost skokového přemístění (teleportace) návštěvníka v rámci modelu, možnost zobrazit si k různým objektům modelu doplňující data (textová, obrazová, ale i třeba zvuková) nebo dokonce s těmito objekty manipulovat.

1.2.2 Obecné požadavky kladené na virtuální model

1.2.2.1 Geometrická přesnost modelu

Výslednou geometrickou přesnost modelu určíme podle účelu, k němuž bude model sloužit. Je zřejmé, že například pro webovou prezentaci památkového objektu není nutno mít model zhotoven

s centimetrovou přesností.

Z praktických zkušeností s tvorbou prostorového modelu vyplývá, že s rostoucími požadavky na přesnost velmi prudce vzrůstá jeho pracnost (viz např. [3]). Přesnější modelování totiž předpokládá zaměřit mnohem více podrobných bodů a konstrukčních měr při geodetických pracech a totéž velké množství dat následně zpracovat v CAD systému. Nesmíme také zapomínat na to, že vysoká požadovaná přesnost zabraňuje použít jednoduché aproximace obecných tvarů (např. nahrazení obecné plochy trojúhelníkovou sítí), což může snížit rychlost zobrazování virtuálního modelu.

Z těchto důvodů je rozumné si požadovanou přesnost modelu dobře rozmyslet a nestanovovat ji větší, než je bezpodmínečně nutné pro potřeby uživatele; při hledání kompromisu mezi přesností modelu a náročností jeho zpracování je třeba postupovat s obezřetností.

1.2.2.2 Vizuální věrnost modelu

Chceme-li vytvořit virtuální model skutečné budovy, jako je tomu v případě krumlovského barokního divadla, napadne nás intuitivně, že by měl mít vzhled maximálně realistický. K tomu můžeme podotknout něco podobného jako ke geometrické přesnosti v předchozím odstavci: praktická zkušenost nás varuje před přehnanými požadavky.

Především musíme znovu vzít ohled na pracnost tvorby modelu. Pro dosažení nejvyšší realističnosti se používají fotogrammetrické metody, kdy se na plochy modelu mapují skutečné snímky pořízené v terénu. Výsledek je velmi efektní, ale v praxi to znamená pořídit dostatečně kvalitní snímky, transformovat je do roviny plochy (někdy i za pomoci zvlášť zaměřených vlíčovacích bodů), barevně je upravit (zejména kvůli světelným přechodům mezi jednotlivými snímky - tzv. barevná ekvalizace snímků) a osadit na příslušné místo modelu. V případě povrchů jako je dlažba, sgrafita ap. si můžeme poněkud ulehčit tím, že opakující se vzor vyfotografujeme pouze jednou a naneseeme jej jako pravidelnou texturu. Obecně je ale vhodné si rozmyslet, zda je fotorealističnost skutečně nutná a zda by u dané plochy nepostačilo definovat pouze barvu povrchu.

Jakákoliv grafická informace ve formě rastru navíc citelně zvyšuje náročnost výpočtů při zobrazování modelu; velké množství fotorealistických ploch tedy může zpomalit práci s modelem v reálném čase.

Na závěr bude ještě vhodné poznamenat, že dokonale reálný vzhled modelu nemusí vždy napomáhat jeho přehlednosti. V některých případech je schematické vykreslení s povrchy v základních barvách mnohem názornější než sice efektní, ale nepřehledné realistické vyobrazení. Na druhou stranu ale v některých případech lze pomocí podrobnější rovinné textury napodobit různé

nerovnosti na povrchu tělesa, aniž bychom je museli modelovat ve 3D.

1.2.2.3 Zobrazování modelu v reálném čase

Uživatel pracující s virtuálním modelem si pochopitelně bude přát, aby se model zobrazoval plynule a co možná nejrychleji. K tomu je třeba maximálně snížit objem výpočtů nutných k zobrazení modelu; ve webovém prostředí, kdy se soubor s modelem přenáší po síti, hraje též velkou roli celkový objem dat a způsob, kterým jsou přenášena. Uveďme si nejběžnější způsoby, jak práci s virtuálním modelem urychlit:

Zjednodušení geometrie modelu: komplikovanost modelu výrazně ovlivníme tím, jakým způsobem jej zkonstruujeme. Model by měl obsahovat co možná nejméně obecných ploch (b-spline plochy ap.) a naopak maximum rovinných prvků. Další možností je použít různé optimalizátory neboli programy, které podle různých algoritmů snižují počet polygonů v modelu.

Zjednodušení povrchových materiálů: model by měl obsahovat pouze nutné minimum rastrových obrázků. Většina povrchů by měla být osazena buď pravidelnými texturami z opakujících se vzorů (ornamentů) nebo prostými barevnými plochami.

Odstupňovaná podrobnost modelu: při pohledu z určité vzdálenosti lze složitější tělesa nahradit jednoduššími, takže zobrazovací výpočty vzdálenějších těles se pak provádějí pro jednoduché geometrické tvary. Pro tento způsob zobrazování objektů se vžil název LOD (Level Of Detail).

Odstupňovaná podrobnost textur (mip-mapping): princip je obdobný jako u LOD; textura je vytvořena v několika různě detailních verzích, mezi nimiž se přepíná v závislosti na vzdálenosti objektu od návštěvníka.

Výpočet zobrazení pouze pro viditelné objekty: do zobrazovacích výpočtů vstupuje každá část modelu teprve ve chvíli, kdy ji návštěvník může vidět; není tedy nutno načítat při návštěvníkově vstupu celý model naráz.

Omezená dohlednost: ve virtuálních modelech se často používá tzv. mlha, která způsobuje, že v určité vzdálenosti od pozorovatele (návštěvníka) začnou objekty postupně mizet a při překročení maximální stanovené vzdálenosti se ztratí úplně. Kromě úspory zobrazovacích výpočtů (neviditelné objekty nejsou do výpočtů zahrnuty) napomáhá mlha i přirozenému prostorovému dojmu.

1.2.2.4 Přehlednost a intuitivní ovládání modelu

Zkušenost ukazuje, že mezi naším pozorováním v reálném světě a pozorováním virtuálního světa pomocí „kamery“ v grafickém softwaru je dosti značný rozdíl. Trojrozměrný model budovy nemůžeme považovat za jakousi „vylepšenou mapu“, totiž plán doplněný o třetí rozměr. Procházet skutečnou budovou a orientovat se při tom podle 2D plánu je mnohem snažší než procházet jejím virtuálním modelem. Prostorový model sám o sobě je sice efektní, ale nepříliš prakticky použitelný. Potřebuje totiž doplnit o dvě důležité pomůcky: o **obslužnou aplikaci**, která umožní jednoduché ovládání pohledu a návštěvníkova pohybu a dále o **navigátor**, pomocí něhož budeme mít v každém okamžiku přehled o tom, kde se v rámci celého modelu nacházíme.

1.3 Volba výchozího a cílového formátu konverze

V této kapitole zabývám především otázkou cílového formátu konverze; popíši stručně nejrozšířenější formáty určené pro virtuální realitu a zdůvodním volbu formátu VRML. Popis formátů bude omezen na to nejnutnější, podrobnější informace čtenář nalezne například v [5]. O volbě výchozího formátu již byla řeč v kapitole 1.1.2, ale přesto se k ní ještě několika slovy vrátím.

V následujících stručných poznámkách se věnuji pouze těm vlastnostem formátů, kterými se od sebe výrazněji liší. Nebudu zmiňovat vlastnosti, které se u virtuálních modelů dají považovat za samozřejmé a všechny uvedené formáty je proto mají – například možnost vybavit model interaktivními prvky.

1.3.1 Možnosti prezentace virtuální reality v počítačových sítích

1.3.1.1 Formát VET (View Experience Technology)

Produkt firmy *Viewpoint Corporation*. Nástroje pro tvorbu, editaci a optimalizaci virtuálních světů ve formátu VET jsou k dispozici z valné většiny zdarma.

Velkou výhodou formátu VET je vysoká realističnost materiálového zobrazení – formát umožňuje realizovat nerovnosti povrchu a pomocí tzv. odrazové mapy dokáže simulovat světelné odrazy. Formát je také velmi úsporný, pokud jde o objem dat – ve srovnání s jinými formáty jsou

soubory VET modelu i třikrát menší (mluvíme-li o případě, kdy model není nijak dodatečně komprimován a neobsahuje žádné textury ve formě rastrových obrázků).

Jeho značnou slabostí ovšem je, že neposkytuje žádné nástroje k vytvoření realistického osvětlení scény. Neumožňuje ani použití mlhy, tj. postupného snižování viditelnosti s rostoucí vzdáleností, které může výrazně napomoci subjektivnímu prostorovému dojmu z modelu. Dále v něm zcela chybí nástroje pro detekci kolizí při průchodu modelem a další čidla (dotyku, přítomnosti návštěvníka. Návštěvník ani nemá možnost manipulovat s objekty modelu. Též není možné se v rámci virtuálního světa “teleportovat” na jiné místo. Formát nezná ani stupňovité zobrazování detailů (LOD). Určitou praktickou nevýhodou může být fakt, že obsahuje-li model textury ve formě rastrových obrázků, jsou tyto obrázkové soubory zahrnuty přímo do souboru s modelem (v interním formátu MTS); změna textury díky tomu znamená editaci modelu ve speciálním softwaru namísto prosté záměny obrázkového souboru.

1.3.1.2 Formát Cult3D

Tvůrcem tohoto formátu je švédská firma CYCORE. Podobně jako u formátu VET, i zde je k dispozici celá řada nástrojů pro tvorbu a editaci modelů, i když plné verze některých z nich již nejsou freeware.

Z našeho hlediska převažují u formátu Cult3D nevýhody nad výhodami. Formát má prakticky tytéž nedostatky, které jsme uvedli u VET. K nim se přidružuje daleko menší úspornost dat.

1.3.1.3 Formát VRML (Virtual Reality Modelling Language)

Je to nejrozšířenější formát pro virtuální realitu. Jeho vývojem se zabývá *Web3D Consortium, Inc.*. Z důvodu velkého rozšíření formátu jsou k dispozici nástroje pro jeho editaci od mnoha různých výrobců, i když jejich plné verze jsou zpravidla licencovány.

Zásadní výhodou VRML je naprostá otevřenost jeho kódu – VRML soubory jsou prostými textovými soubory, které jsou pouze kvůli rychlejšímu přenosu po sítích komprimovány běžným programem *gzip*. To poskytuje velké možnosti především programátorům vyvíjejícím aplikace, které s tímto formátem pracují.

Oproti výše uvedeným formátům má několik nedostatků, především pokud jde o realističnost modelu: neumožňuje modelovat nerovnost povrchu, nedokáže zobrazit světelné odrazy na povrchu těles. Naopak ale umožňuje vložit do modelu několik druhů virtuálních světél a dokáže pracovat s

mlhou, což k realističnosti modelu naopak významně přispívá. Dále obsahuje řadu důležitých a praktických nástrojů, jako jsou čidla dotyku, viditelnosti a přítomnosti návštěvníka, detekce kolizí, nástroje pro manipulaci s objekty modelu při jeho procházení, teleportace na jiná místa modelu. Praktická je i práce s texturami uloženými v obrázkových souborech: soubory s texturami nejsou součástí souboru v interním formátu, nýbrž zůstávají zvlášť a zdrojový kód VRML se na ně pouze odkazuje.

1.3.1.4 Formát X3D

Formát X3D je podobně jako VRML produktem společnosti *Web3D Consortium, Inc.* a má se v dohledné době stát jeho nástupcem. Formáty budou kompatibilní v tom smyslu, že VRML bude pomnožinou X3D; z toho důvodu se v dnešní době nemusíme nijak obávat práce s formátem VRML. Vylepšení by mělo spočívat především v oblasti interaktivity virtuálních modelů. Pro podrobnější informace o vývoji X3D odkazují na [6].

1.3.2 Volba VRML jako cílového formátu

Jako cílový formát konverze jsem zvolil formát VRML; zde uvádím důvody, které mě k tomu vedly.

Otevřený zdrojový kód: je-li cílem vytvořit poměrně složitou aplikaci, jejímž produktem bude soubor s virtuálním modelem, je třeba nejen přesně znát způsob, kterým daný formát pracuje, ale musíme také být schopni jeho zdrojový kód fyzicky vytvořit. Otevřený formát nám kromě toho umožňuje zajistit bezchybnou kompatibilitu různých aplikací a jejich verzí.

Dobrá strukturovanost formátu: formát je velmi přehledně uspořádán; vyhledávání a manipulace s daty, ať už automatizovaná nebo “ruční”, je intuitivní.

Přítomnost interaktivních prvků: k interaktivitě a zajímavosti modelu výrazně přispívají **čidla**, která umožňují například spuštění dynamické události ve chvíli, kdy se návštěvník přiblíží k nějakému objektu, “dotkne” se ho nebo dokonce když ho pouze spatří. Dalším výhodnou vlastností VRML je možnost **teleportace v rámci modelu:** jestliže lze návštěvníka přemísťovat skokem mezi různými stanovišti, znamená to možnost řízené prohlídky objektu, což u těch modelů, které budou sloužit například k propagaci památkového objektu, uživatelé ocení.

Model ve více souborech: rozdělení virtuálního modelu do více souborů výrazně přispívá k jeho

přehlednosti a navíc poskytuje možnost urychlit práci s modelem ve webovém prostředí (data modelu se přenáší po částech).

1.3.3 Volba DXF jako výchozího formátu

O volbě DXF jako vstupního formátu konverze již byla řeč v kapitole 1.1.2. Důvody volby jsou zřejmé: DXF je formát **otevřený** a celosvětově **rozšířený**.

Otevřenost formátu zajišťuje, že budeme schopni vyvinout takovou aplikaci, která dokáže velmi jednoduchým způsobem pracovat s daty ze zdrojového souboru (DXF výkresu); DXF výkres totiž není nic jiného než prostý textový soubor.

Univerzalitou DXF máme na mysli to, že se tento formát stal standardem de facto v oblasti vektorových dat. Prakticky všechny rozšířené CAD systémy s ním dokážou pracovat, a proto i možnosti využití navržené konverzní aplikace budou širší.

1.4 Volba platformy pro tvorbu konverzního programu

Tato kapitola se zabývá základními rysy konverzního programu, který je mým cílem navrhnout. Nejprve zdůvodním, proč považuji za výhodnější pojmout program jako nadstavbu nějakého CAD systému namísto samostatné aplikace; jako vhodnou platformu poté zvolím software AutoCAD. Na závěr se krátce zmíním o nejpoužívanějších programovacích nástrojích, které vývojář aplikací tohoto typu v AutoCADu má.

1.4.1 Samostatná aplikace nebo nadstavba CAD systému?

Je jistě možné představit si konverzní program jako zcela samostatnou aplikaci, která by načetla vstupní DXF soubor a umožnila uživateli převést jej na formát VRML. To by ovšem bylo praktické pouze v tom případě, kdy by konverze probíhala zcela automaticky; v momentě, kdy proces není kompletně automatizován a uživatel chce do tvorby VRML modelu jakkoli zasahovat (např. seskupit některé prvky do větších objektů, nastavit u různých prvků různou kvalitu hladkosti povrchu nebo konvertovat pouze vybranou část modelu), je nezbytně nutné, aby měl možnost provést výběr z DXF a zároveň aby měl okamžitý intuitivní náhled na objekty, s nimiž momentálně zachází. Jinak řečeno, uživatel potřebuje zobrazit DXF soubor v nějaké aplikaci, která s ním dokáže

rozumně pracovat.

Je zřejmé, že vytvářet takovou aplikaci jako součást konvertoru je zbytečné, jestliže podobné aplikace – totiž CAD systémy – již existují. Z toho důvodu je mnohem jednodušší opačný přístup: vytvořit konvertor jako součást (nadstavbu) nějakého CAD systému. Nadstavba může se samotným systémem velmi účinně spolupracovat: je-li třeba vybrat některé prvky z DXF, děje se tak způsobem běžným v daném CADu; jestliže se v konvertoru manipuluje s nějakým prvkem, může se tento prvek v DXF zvýraznit apod.

Vytvářet konvertor jako nadstavbu CADu má ještě jednu podstatnou výhodu z hlediska programátorského, že totiž lze využít programovací nástroje uzpůsobené přímo pro vývoj CAD aplikací. Triviální příklad: chceme-li vyhledat v DXF souboru nějaký konkrétní prvek, nemusíme vytvářet algoritmy procházející slovo po slově celý DXF soubor a hledající příslušný řetězec znaků nebo příslušné číselné hodnoty, nýbrž použijeme připravenou proceduru (funkci), která prohledá databázi výkresu. Nejvýrazněji se výhody tohoto přístupu projeví v objektově orientovaných programovacích jazycích, které využívají tzv. objektový model daného CAD systému; pak máme v ruce nástroj doslova “šitý na míru”.

1.4.2 Volba vhodného CAD systému

Rozhodl jsem se tedy vytvořit konvertor jako nadstavbu některého CAD systému. Podmínkou je, aby tento systém dokázal pracovat s 3D výkresy ve formátu DXF (to je zajištěno u většiny CADů) a aby poskytoval možnost tvorby vlastních nadstaveb. V neposlední řadě je pak důležité, aby byl co nejvíce rozšířen – v opačném případě by se konverzní aplikace stala programem exkluzivním a málo využitelným.

Instinktivně se nabízí software firmy **Autodesk**, která vyvinula samotný formát DXF a jejíž produkty patří v oblasti CAD a vizualizace mezi nejrozšířenější na světě. Z nich pak co do rozšíření padá volba jednoznačně na **AutoCAD**. Ostatní software tohoto výrobce (Autodesk VIZ, ArchiCAD...) lze považovat za příliš specializovaný; našim potřebám bohatě postačí aplikace, která bude schopná zobrazit a jednoduše editovat formát DXF. Zdůrazňuji, že hledáme software pro *otevření* DXF modelu a jeho *konverzi*, nikoli pro *konstrukci* modelu. Ten je možno vytvořit v jakémkoli jiném systému a AutoCAD posléze použít pouze jako “přestupní stanici” pro konverzi do VRML.

1.4.3 Krátce o tvorbě nadstavbových aplikací v AutoCADu

Nejjednodušší programovací nástroj AutoCADu, prostředí **Visual LISP**, je ideálním nástrojem pro tvorbu a ladění menších aplikací; jazyk je jednoduchý a umožňuje poměrně snadnou práci se základními funkcemi AutoCADu. Vývojové prostředí je dodáváno jako integrovaná součást AutoCADu. Při vývoji komplikovanějších aplikací může ovšem být na překážku nepříliš přehledný zápis, absence knihoven a v neposlední řadě dosti slabá sémantická kontrola. Spolupráce AutoCADu s jinými aplikacemi je sice ve Visual LISPu možná, ale poměrně neohrabaná. Obecně tedy lze říci, že jazyk Visual LISP je vhodný pro tvorbu menších aplikací (funkcí) pracujících v rámci AutoCADu samotného.

Mnohem mocnější nástroj představuje vývojové prostředí **VBA (Visual Basic for Applications)**. Toto objektově orientované prostředí je integrováno do různých aplikací běžících pod systémem Windows (MS Word, MS Excel, AutoCAD...), které je možno přímo z VBA řídit pomocí rozhraní *ActiveX Automation*. Toto rozhraní je – zjednodušeně řečeno – nástrojem, který komunikuje “na jedné straně” s libovolnou aplikací, která ho podporuje (Word, AutoCAD...) a “na druhé straně” s prostředím VBA. Díky tomuto rozhraní je vlastně možné vytvořit program, který bude využívat služeb několika aplikací zároveň (příkladem může být kreslení v AutoCADu na základě dat zadávaných do MS Excelu). Jazyk je jednoduchý a uschopňuje programátora k velice efektivnímu řízení hostitelské aplikace. Vývojové prostředí, které ji dodáváno spolu s AutoCADem, navíc nabízí možnost velmi rychlého návrhu dialogových oken a jednoduchého doplnění zdrojového kódu jednotlivých jejích prvků.

Podstatně větší možnosti – ovšem za cenu použití složitějšího jazyka – nabízí programovací prostředí **ObjectARX (ObjectAutoCAD Runtime Extension)**, které obstarává propojení AutoCADu a jazyka C++. Je to plně profesionální nástroj, který právě díky spolupráci s jazykem C++ umožňuje vytvořit libovolně náročnou aplikaci pracující s AutoCADem nebo jinou aplikací. Umožňuje přímý přístup k funkcím AutoCADu i do paměti, takže aplikace vytvořené v C++ pracují rychleji než v jiných programovacích jazycích.

Pro podrobnější informace doporučuji nahlédnout do [12].

1.5 Postup další práce

Zde je krátké shrnutí, k čemu jsem dospěl v této kapitole: cílem je navrhnout nadstavbovou aplikaci AutoCADu, která dokáže poloautomaticky konvertovat prostorový DXF soubor na model ve formátu VRML, a dále přizpůsobit stávající model krumlovského barokního divadla tak, aby se mohl stát vstupním souborem této konverze.

Další postup práce bude následující:

- (a) sestavení pravidel, podle nichž má být tvořen DXF model (kapitola 2),**
- (b) úprava modelu barokního divadla podle těchto pravidel (komentář v kapitole 2.5),**
- (c) stanovení struktury výsledného VRML modelu (kapitola 3),**
- (d) návrh konverzní aplikace (kapitola 4).**