

## Využití prostředků virtuální reality pro zobrazení technologického procesu hlubinného dolu

Oldřich Kodym<sup>1</sup>

### Virtual Reality in Presentation of the Underground Mine Technological Process

*Virtual Reality in Presentation of the Underground Mine Technological Process focuses on methods of presentation of an underground mine technologies in intranet technology. It shows usage of platform independent VRML client for presentation of static and dynamic information about technological process. Bi-directional interactions between client and process information database are solved.*

*Based on analysis of technological process of underground mine a database structure was designed. It is skeleton for storing all information about any underground mine. This skeleton can be modified in any direction. Data in this "static model" of underground mine can be applied for visualization in VRML environment. In this way it is possible to simplify and unify a user's front-end for all kinds of tasks.*

*All designed scenes can be interactively displayed in full view or in any detail view, so that a user is able to recognize every important part of installed equipment, its stage, technical parameters and other information. If manufacturers of mining equipment will supply VRML model of their real products everybody would be able to place it into VRML scene and learn everything about it.*

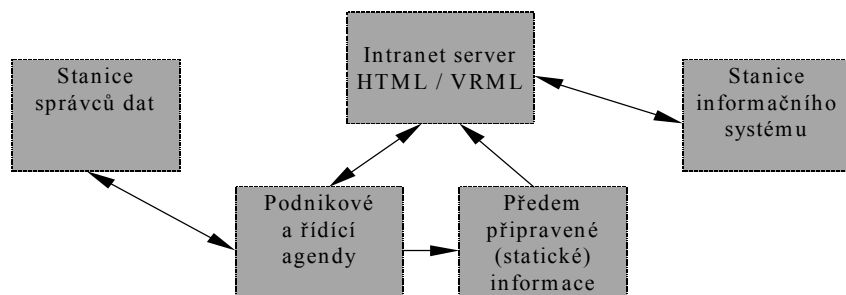
*This work explores and tries to enlighten some of the areas and available approaches compliant with VRML 97 specification of modifying static scene by its browser. Concepts of animation pipeline, inside and outside scripting in scene displayed and authoring of VRML targeted geometry are discussed including database connectivity.*

**Key words:** model, visualization, virtual reality, underground mine, 3D.

### Úvod - moderní technologie pro práci s informacemi

Rozvoj informačních technologií umožňuje vytvářet se prostředí, kde lze předkládat informace strukturované tak, že mohou být velice snadno vyhledány i zobrazeny, i když jde o informace zcela odlišného typu. Přístup k dalším rozšiřujícím informacím, dokumentům a pod. lze řešit pomocí odkazů.

Tato technologie ulehčuje uživatelům práci s informacemi v tom, že jim odpadá starost o to kde a jak jsou požadované informace uloženy. Všechny tyto informace jsou ukryty právě ve vhodně definovaném odkazu. Na obr. 1. jsou zobrazeny hlavní části informačního systému s datovými toky, které prezentují jednak požávku po informacích a cesty jejího naplnění. Důležitou součástí je i správa a údržba dat i celého systému.



Obr.1. Základní komponenty informačního systému, založeného na Intranet technologii.

Fig.1. Basic components of information system, based on the Intranet technology.

Vzájemnou komunikaci, týmovou práci a společný přístup k informacím umožňuje každá podniková síť. Důležitou částí návrhu podnikového Intranetu je skloubení různých zdrojů informací. Nejsnazším řešením je příprava všech nabízených informací ve formě HTML dokumentů, které jsou na klientských stanicích informačního systému pouze pasivně zobrazeny. Samozřejmou součástí těchto dokumentů jsou příslušné hyperlinky na navazující nebo doplňující dokumenty jiné nebo jiné části aktuálního dokumentu.

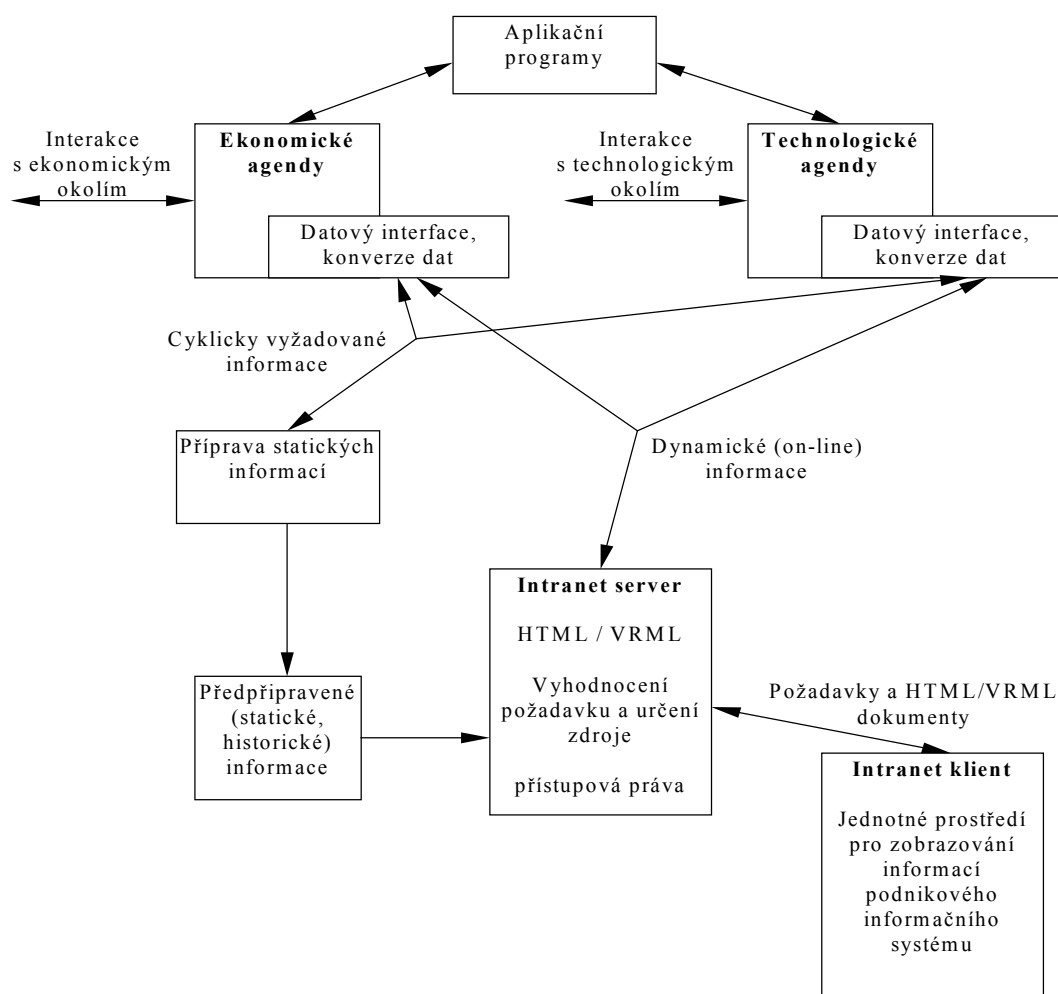
Pro integraci informací z technologického procesu je nutno volit jiné postupy, které by zajistily maximální aktuálnost prezentovaných informací. Právě v oblasti vizualizace informací z technologického procesu vynikne další vlastnost Intranet technologie - dokumenty ve formátu VRML, které popisují 3D scénu a s vhodným prohlížečem umožňují i interaktivní pohyb v této scéně. Ne každý vizualizační software však má tyto grafické prezentační možnosti, takže v této oblasti se obě technologie mohou výhodně doplňovat. Samozřejmou vlastností 3D scény je její interaktivnost, i sem je možné umisťovat odkazy na detailnější resp. rozšiřující

<sup>1</sup> Dr. Ing. Oldřich Kodym: Vysoká škola báňská – TU Ostrava, Hornicko-geologická fakulta, Institut ekonomiky a systémů řízení, Oddělení automatizace v hornictví. 17. listopadu 15, 708 33 Ostrava – Poruba, Česká Republika e-mail: oldrich.kodym@vsb.cz (Recenzovaná a revidovaná verzia dodaná 3.5.2003)

informace. Interaktivita rovněž přináší možnost opačného toku dat (z 3D scény přes datové úložiště až do technologického procesu) pro účely ovládání i řízení (Kodym, 1997).

### Nezbytné softwarové prostředky

Základními komponenty tohoto klient/server informačního systému je WWW server, který zpracovává požadavky klientů a zasílá jim vyžádané HTML nebo VRML dokumenty a tzv. prohlížeč (browser), který zobrazuje vyžádané dokumenty. Komponenty jsou tedy naprosto shodné s Internetem, což přináší některé výhody. WWW servery i klienti jsou k dispozici téměř pro všechny platformy jak hardwarové (od osobních počítačů přes pracovní stanice až po třídu mainframe) tak i softwarové (Windows WKG - NT, OS/2, OS7, UNIX, Netware a další). To umožňuje prohlížet stejné dokumenty na různých počítačích s různými operačními systémy; do podnikového informačního systému lze tedy zařadit jakékoliv existující počítače, které mohou být primárně určeny k řešení jiných úkolů či agend. Rovněž masové nasazování klientů snižuje jejich pořizovací náklady - některé z nich jsou k dispozici buď zdarma nebo jsou přímo součástí základní konfigurace operačního systému.



Obr.2. Základní datové interakce.  
Fig.2. Basic data interactions.

### Informační zdroje

Nejcennější vlastností informačního systému jsou informace v něm obsažené, jejich správnost, přesnost a aktuálnost. Z tohoto hlediska můžeme informace rozdělit do skupin podle časového horizontu v němž splňují výše uvedená kritéria. S tímto rovněž souvisí způsob jejich uložení pro zobrazení v informačním systému.

Informace, které se v čase nemění a jejichž hodnota je stálá, je vhodné uložit do běžných HTML nebo VRML dokumentů, které jsou neustále k dispozici na Intranet serveru. Tyto dokumentace jsou vytvořeny jednou, obvykle v okamžiku, kdy jsou k dispozici všechny potřebné informace buď automatizovaně nebo ručně.

Do této kategorie lze zařadit informace typu popis výrobku, výroční zpráva, ekonomické ukazatele za uplynulé období a pod.

Jiným problémem je prezentace dat z technologického procesu, které jsou proměnné v čase v podstatně kratších časových horizontech. Je-li zapotřebí jejich zobrazení, musí být jejich aktualizace prováděna v čase srovnatelném s rychlostí jejich změn. Bylo by principiálně možné použít výše uvedený postup, ale toto řešení je pro potřeby podnikového informačního systému nedostatečné, protože neúměrně roste četnost přípravy „statických“ dokumentů a v některých aplikacích vnáší nepřijatelné zpoždění (=neaktuálnost) informací.

Mnohé vyžadované informace je třeba prezentovat téměř v režimu on-line, s minimalizovaným časovým zpožděním. Zde je možnost integrovat do provozovaných dispečerských systémů moduly, které na vyžádání z informačního systému vygenerují buď přímo dokument v HTML formátu nebo alespoň data, na základě kterých lze dokument vytvořit dodatečně (podpůrným účelovým programem) a okamžitě jej odeslat k zobrazení. Server má totiž možnost spouštět programy na základě požadavků od prohlížečů (klientů), které mohou odeslat dotaz formulovaný v žádaném jazyku (např. do SQL databáze nebo dispečerského systému) a převzít data nebo přímo vygenerovaný dokument. Základní datové interakce jsou uvedeny na obr. 2.

Pro realistické zobrazování technologických procesů jsou využívány dokumenty ve formátu VRML, které popisují 3D scénu včetně okamžitých procesních informací. Využitím interaktivnosti scény je možné realizovat odkazy na detailnější resp. rozšiřující informace.

Jak zde bylo naznačeno, možnosti využití Intranet technologie jsou i v oblasti, která je dosud vyhrazena účelově orientovaným programovým systémům. Vzhledem k perspektivě, kterou autor této technologii dává, lze předpokládat, že „front end“ části jednotlivých systémů se budou postupně sjednocovat a unifikace tohoto rozhraní může zvýšit ergonomii práce.

### Prostředí VRML pro realizaci 3D technologické scény

Pro popis 3D scény lze využít různých postupů. Mnohé systémy využívají svůj vlastní, účelový jazyk, jehož formát nebývá zveřejňován, a které tedy umožňují pouze vizualizaci scény v konkrétním systému. Na druhé straně výsledkem snah o unifikaci a vytvoření univerzálního přenositelného jazyka se stal standard VRML dnes ve verzi VRML97. Vychází z podobné filozofie jako jazyk HTML, který je běžně používán pro prezentaci dat v prostředí WWW. Zjednodušeně na něj lze pohlížet jako na rozšíření 2D dokumentů (popisovaných jazykem HTML) na plně 3D dokumenty (popisované jazykem VRML).

#### Vlastnosti jazyka

Jazyk VRML je charakterizován podobně jako HTML jako plně hypertextový popisovač 3D scény. Mezi jeho hlavní charakteristiky patří:

- textový formát,
- jednoduchý popis 3D objektů,
- standard prostředí WWW,
- podpora dodavateli grafických subsystémů a WWW prohlížečů.

Prvky jazyka jsou uzly, pole, hodnoty atd., které popisují 3D scénu. Základními objekty jsou hranol, koule, kužel a válec. Z těchto objektů je možno sestavovat komplexnější tvary. Každý objekt je definován svými rozměry, umístěním a materiálem, který určuje jeho vzhled. Při definici vlastních objektů se využívá postupů strukturovaného programování pro sdílení různých vlastností. Podrobnější informace jsou uvedeny ve specializovaných publikacích včetně programových skriptů pro realizaci v jazyce neobsažených funkcí.

#### Funkční rozhraní prohlížečů

Pro konečnou vizualizaci virtuální technologické scény slouží specializované programové produkty, které interpretují zdrojový text popisující tuto scénu. Výhodou těchto VRML prohlížečů je, stejně jako u HTML prohlížečů, to, že jsou k dispozici pro různé platformy počítačů.

Základní samozřejmou funkcí prohlížeče je zobrazení 3D technologické scény na obrazovce monitoru. Pomocí ovládacích prvků se lze potom v dané technologické scéně pohybovat. K dispozici je pohyb pozorovatele vůči scéně s možnostmi pohyb vpřed, vzad a otáčení, posun v horizontálním a vertikálním směru a otáčení pozorovatele. Dále lze pohybovat scénou vůči pozorovateli, opět s možnostmi posunu, pohybu a otáčení.

Pro ovládání prohlížeče je možné využít několika různých zařízení. Nejčastěji se používá klasická klávesnice, myš nebo joystick. Požadavky na počítač vyplývají z toho, že se provozuje aplikace provádějící plně 3D zobrazování v reálném čase, což znamená, že jak procesor tak i grafický subsystém musejí být (na platformě PC) z horního konce výkonového spektra. Příklad realizované 3D scény povrchových provozů je na obr. 3.



Obr.3. Zobrazení povrchových provozů dolu.  
Fig.3. Depiction of surface mine operations.

### Realizace rozšířených funkcí

Skriptování je mechanismus dovolující do scény zapojit logiku vyšších programovacích jazyků. Vychází ze struktury prohlížeče (viz. obr. 4.). Lze je realizovat dvěma způsoby: v rámci zdrojového textu scény (viz. obr. 5.) a z vnějšku prohlížeče – z okolí (viz. obr. 6.).

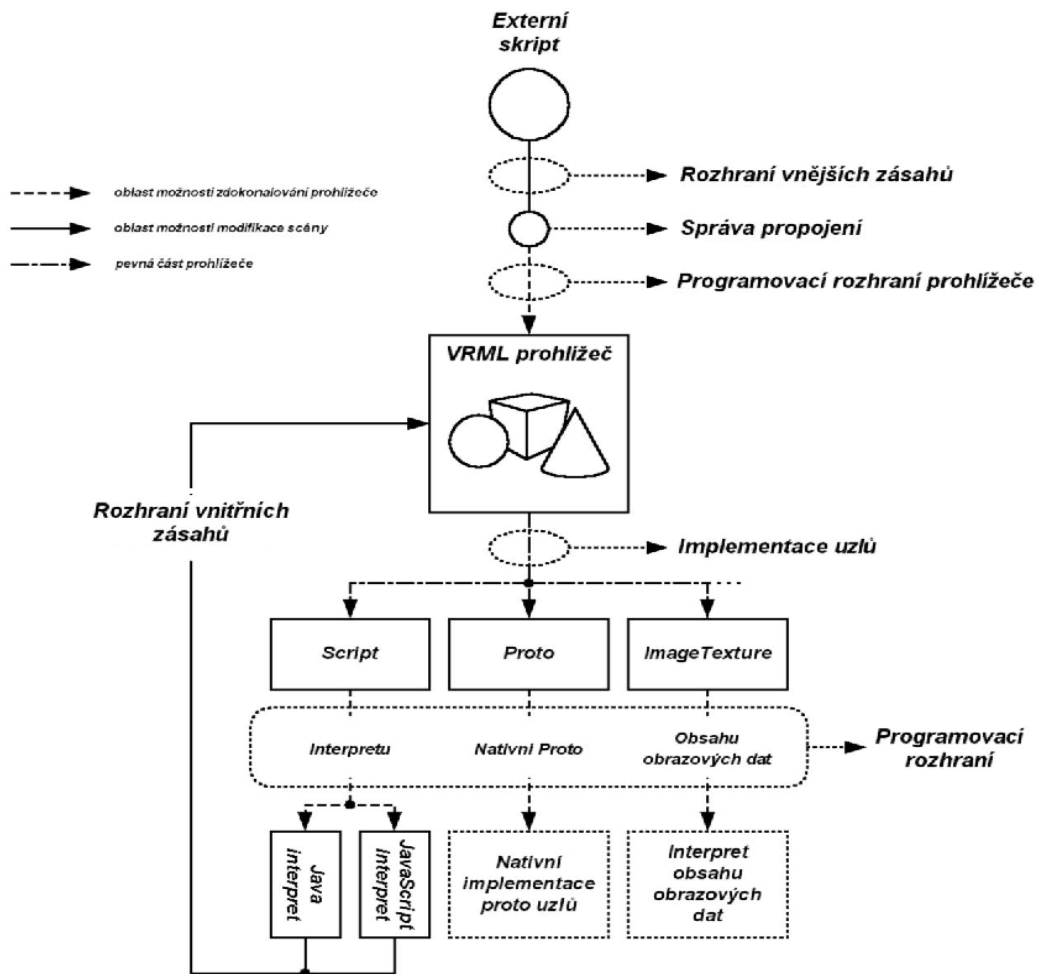
Druhým zdrojem ovlivňování chování scény je externí skript, který se na prohlížeč napojuje prostřednictvím standardního rozhraní vnějších zásahů EAI, založeném na programovacím rozhraní prohlížeče. Na něj navazuje samotná realizace jednotlivých funkčních modulů, implementace uzlů. Ideálním způsobem návrhu prohlížeče je jeho modularita umožňující rozšiřování funkcí. Modularita je zde chápána jako definice programového rozhraní pro implementaci vybraných uzlů na jejichž základě lze vytvářet knihovny pro specifická zpracovávaná data. Například pro uzly typu script jsou realizovány jednotlivé interprety skriptovacích jazyků (Java, ECMA skripty, Java skripty). Tyto interprety v rámci scény pak mohou zpětně ovlivňovat její obsah přes rozhraní vnitřních zásahů.

Propojení VRML scény s databázovým zdrojem geometrie lze volit ze dvou variant: buď přímo ze skriptu právě zobrazované scény (přístup Oracle). Nevýhodou tohoto řešení je závislost na prohlížeči, resp. na zobrazované scéně – všechny podněty za strany uživatele procházejí do skriptu přes právě zobrazovanou scénu. Uživatelské rozhraní zde musí být realizováno přímo pomocí VRML, což je u komplexnějších uzlů ve scéně téměř vyloučeno.

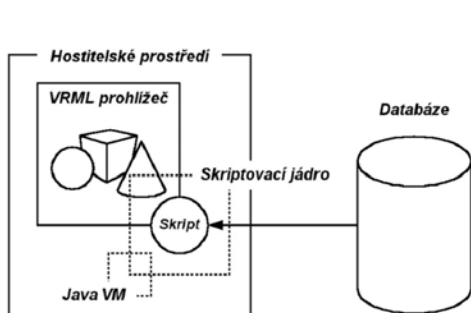
Propojení pomocí externího mezičlánku je druhou možností. Využívá EAI a odstraňuje omezení předchozí varianty. Původně byly pro vnější zdroje zásahu navrženy pouze Java Applety (pro svou nezávislost na použité platformě). Tato nezávislost je velmi důležitá právě v prostředí Intranetu, protože přináší i aplikační nezávislost. Stejně bylo navrhováno rozhraní EAI jako nevázané na prohlížeč konkrétního výrobce. V rámci standardizačního procesu již prošlo zkušebními implementacemi a W3D jej přijalo jako regulérní rozšíření stávající specifikace. Rovněž výrobci nejoblíbenějších prohlížečů jej již integrovali. EAI bylo původně implementováno jako doprovodné Java třídy umožňující Appletu běžícímu společně na jedné HTML stránce s VRML prohlížečem zasahovat do jím zobrazované scény.

Toto řešení je však pro řešení konkrétních zobrazovacích úloh příliš obecné. Aplikace mohou být realizovány pouze jako Applety, což přináší především bezpečnostní omezení. Závažnějším problémem je nestabilní a nestandardizované prostředí v němž tyto Applety pracují. Jde o interakci mezi rozhraním VRML prohlížeče, Appletu v rámci nadřazeného WWW prohlížeče a jeho vlastní Java VM. Každý výrobce realizuje Java VM odlišným způsobem, což ve svém důsledku přináší rozdílné chování s různou mírou stability.

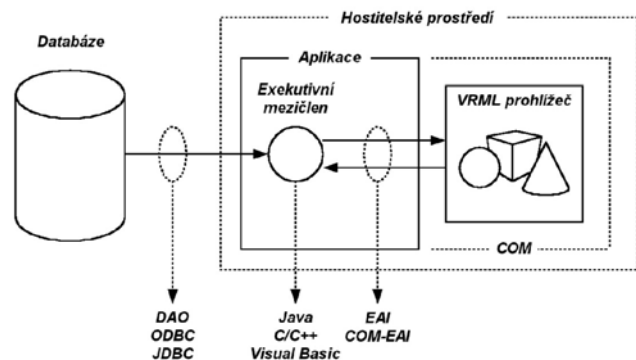
V současné době však byla funkcionalita EAI implementována i jako množina COM rozhraní prohlížeče. Tento přístup není zcela v souladu se záměry W3D. Implementace VRML prohlížeče jako komponenty a tím i pravděpodobná dodatečná realizace EAI pomocí množiny COM rozhraní řeší výše uvedené nedostatky právě možností separace VRML prohlížeče, který se tak stává nezávislým na klientské aplikaci (hostitelském prostředí) a lze jej standardním způsobem zakomponovat do téměř libovolné aplikace vyvinuté v jazyce podporujícím COM technologii. Daní za to je ztráta platformní nezávislosti, protože toto řešení nebylo dosud standardizováno.



Obr.4. Vnitřní struktura prohlížeče.  
Fig.4. Internal structure of the browser.



Obr.5. Skriptování přímo v popisu scény.  
Fig.5. Script directly in the scene.



Obr.6. Skriptování mimo prohlížeč v externí aplikaci.  
Fig.6. Script off the browser in an external application.

Technologie COM vychází z trendů nastolených objektově orientovaným programováním. Neudržitelnost vývoje všeobsahujících monolitických aplikací a potřeby znovupoužití již napsaného kódu, distribuovanost aplikací v síti či mezi procesy vedla ke změnám v programovacích technikách, a tedy v pojetí aplikací. COM není programovacím jazykem, ale jen programovacím modelem a definicí binárního standardu. Na platformě Win32 vyšel z technologie OLE a stal se základem ASP, DirectX, ActiveX a dalších technologií. Je příkladem klient/server technologie, kde aplikace je klientem služeb poskytovaných komponentou (serverem) zprostředkovaných přes implementačně nezávislá rozhraní. Každé rozhraní komponenty dědí (ve smyslu COM

implementuje) implicitní rozhraní IUnknown, které svými třemi metodami zajišťuje životní cyklus komponenty (počet využívaných instancí, orientaci v množině komponentou nabízených rozhraní).

Skriptování v rámci zdrojového textu scény je realizováno pomocí dedikovaného uzlu Script. Poskytuje tvůrcům VRML scény možnost rozšíření množiny dostupných uzlů o uzly uživatelské, ale ve srovnání s technikou prototypování nezávislé na funkcionalitě uzlů standardních. Pro konkrétní script jsou důležité následující položky:

- URL – adresy souborů (jedna nebo více) obsahující zdrojový kód určený pro interpretaci nebo přímo vložený text programu s uvedením protokolu jazyka.
- DirectOutput – určuje, je-li script schopen ovlivňovat strukturu scény přímo přístupem k položkám uzlu i přidáváním resp. odebráním jednotlivých propojení nebo pouze komunikací s prostředím scény pomocí předem definovaných vazeb.
- MustEvaluate – schopnost interpretace daného uzlu i při změně vstupních událostí bez nutnosti explicitně žádat o příslušné hodnoty napojených událostí.
- EventIn, eventOut, Field- určují veřejně přístupné a privátní položky uzlu.

Zásah do zobrazované scény z jejího okolí je, jak již bylo uvedeno dříve, založen na rozhraní vnějších zásahů – EAI. Lze jej realizovat dvěma ekvivalentními způsoby: standardními Java třídami z prostředí Java kódu (Appletu) nebo pomocí technologie COM, kde API EAI je reprezentováno podmnožinou exponovaných a implementovaných komponentou prohlížeče. Podstata realizace je v obou případech obdobná. Aplikace sledují (využitím EAI) posloupnost akcí, jejichž cílem je přístup k určitému uzlu a modifikace jeho parametrů:

- Browser::getBrowser – získání přístupu k VRML prohlížeči a jeho prostřednictvím k zobrazované scéně.
- Browser::getNode, Browser::createVrmlFromString – zpřístupnění nebo vytvoření reference žádaného uzlu. Přístupné jsou pouze ty uzly, které byly ve zdrojovém textu scény definovány a jsou prohlížečem zařazeny v interní tabulce referencí uzlů.
- Node::getEventIn/Out – zpřístupnění vhodné události.
- Event::set/getValue – konečné čtení nebo zápis události určené jménem a datovým typem náležející příslušnému uzlu, který tvoří součást scény.

Toto je velmi zjednodušený popis možností rozhraní EAI. V současné době je technologie COM postupně nahrazována technologií NET, která rozšiřuje možnosti aplikací tím, že již není nezbytné realizovat všechny komponenty aplikace na jednom počítači, ale aplikaci lze distribuovat v rámci počítačové sítě.

## **Modely zařízení a technologií**

Pro navrhování geometrie modelů lze použít mnoha postupů. Jedním z možností je použít např. 3D Studio MAX. Je to jeden z mála prostředků, který je vybaven nativní podporou formátu VRML nejen na straně kvalitního exportu ale i v procesu vlastního navrhování. Již během modelování lze do scény vkládat pomocné geometrické i negeometrické objekty reprezentující VRML uzly. Možnosti exportního filtru jsou velmi široké: od možnosti generování normálových vektorů dílčích polygonů přes generování správných primitiv, barevného určení polygonů i vrcholů, vkládání koordinačních interpolátorů pro potřeby animací, definice kamer a osvětlení až po vkládání negeometrických prvků scény, volbu formátu a přesnosti generovaných hodnot. Výhodnou funkcí je sledování počtu polygonů, na jehož základě lze usuzovat na složitost scény a odhadnout rychlost a náročnost výpočtu a vykreslení v prohlížeči. Ukázky jsou uvedeny na obr. 7. a 8.

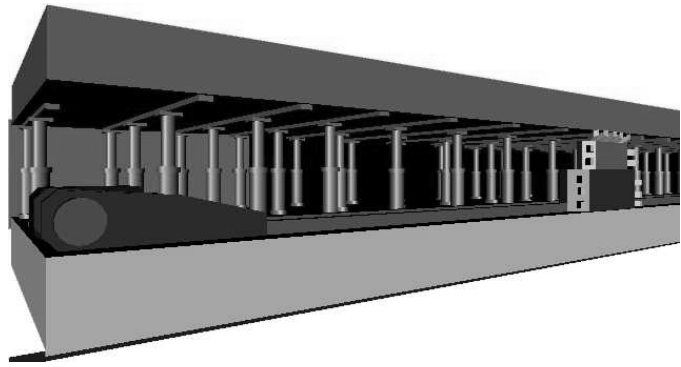
Koncepce uložení geometrie vytvořených modelů v relační databázi mohla být zvolena buď rozložením struktury veškerých uzlů zakotvených daným modelem (transformace, souřadnice vrcholů, materiály atd.) a jejím následným uložením tak, aby bylo možno provést zpětnou jednoznačnou agregaci nebo uchování modelů jako monolitických bloků dat geometrie po jednotlivých celistvých prvcích (technologických celcích). První z uvedených možností předpokládá komplexní návrh databázové struktury pokrývající všechny možnosti syntaxe jazyka VRML. Takto lze zajistit optimální vazbu na scénu, klade však vysoké nároky na návrh a implementaci obslužných algoritmů s implementovanou plnou analýzou VRML kódu a jeho mapování do databáze.

Pro vlastní řešení byl zvolen kompromisní přístup se zaměřením na struktury schopné následného mapování do hierarchie souborového systému. To umožní aplikaci jednoduše exportovat obsah vytvořené scény v podobě „šablonovitého“ VRML zdroje obsahujícího pouze odkazy a transformace vložené z databáze do scény modelů. Pro uložení vlastního popisu geometrie (textu) se předpokládá ANSI SQL typ CLOB s omezením, které přináší jeho konkrétní implementace v prostředí MS Access (typ Memo). Jeho omezený rozsah je postačující pro uchování samotné geometrie, přináší však problémy s připojováním textur.

Samozřejmě existuje možnost realizovat VRML scény v běžném textovém editoru. Tento postup se však při komplexnějších scénách stává nepřehledným a neefektivním. Nicméně byl použit pro úvodní jednoduché neinteraktivní scény.



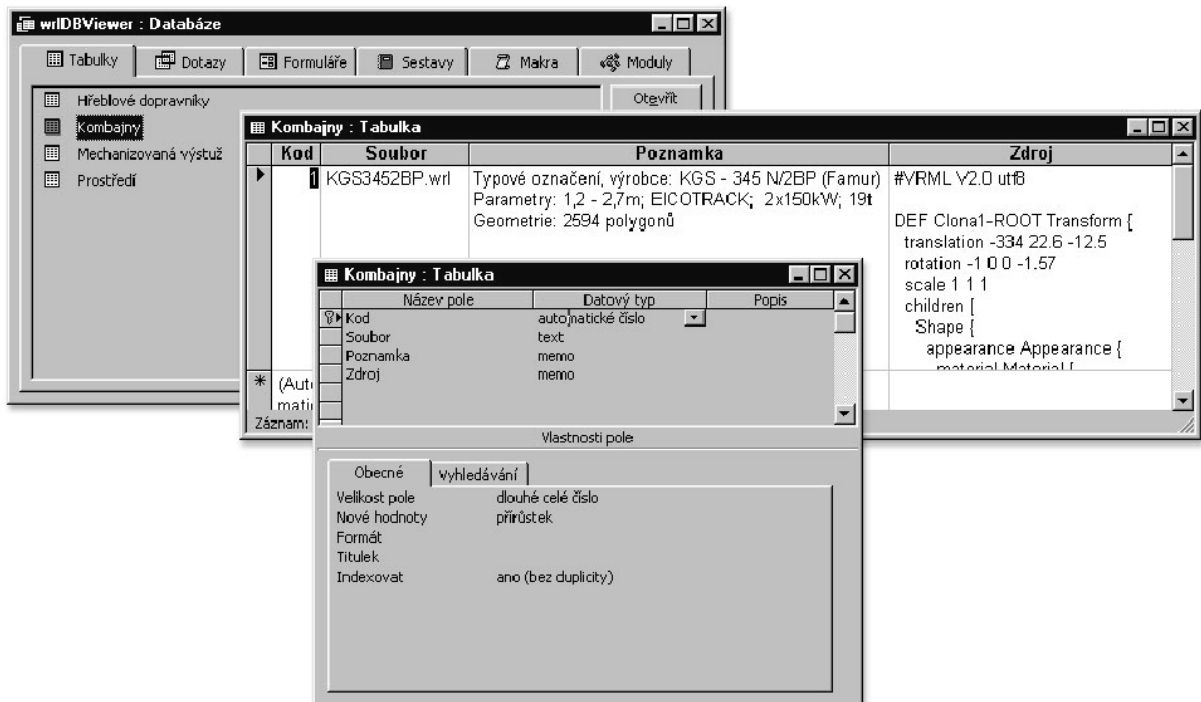
Obr.7. Model hřeblového dopravníku.  
Fig.7. Model of scraper conveyer.



Obr.8. Pluhový porub.  
Fig.8. Plough face.

### Přístup k datům

Databáze je složena z jednotlivých tabulek pro definici určitých prvků technologického celku (Kodym, 1997) (dobývací kombajny, posuvné výztuže, pásové dopravníky atd.). Tyto pak obsahují konkrétní geometrii jednotlivých modelů i s možností popisu (výrobce, typ, charakteristické a provozní parametry atd.) i dalších přidávaných informací (viz. obr. 9).



Obr.9. Realizovaná databáze.  
Fig.9. Realized database.

Jedno z hlavních omezení tohoto způsobu uložení dat v databázi je absence vnějších propojení (externí prototypy, textury, animace, zvuky atd.). Absence textur je velmi citelná, protože tímto způsobem lze velmi účinně rozšířit informaci nesenou zobrazovanou scénou. Použití textury také usnadňuje zobrazování, protože nahrazuje výpočetně náročnou geometrii umístěním vhodné textury, což je realizováno v režii 3D akcelerovaného videoadaptéru. Řešením je využití VRML uzlu PixelImage. Do tohoto uzlu lze uložit nejen texturu ale i její další vlastnosti – zbarvení, průhlednost a barevnou informaci včetně alfakanálu.

Vlastním řešením je aplikace wrIDBViewer, která vychází z výše uvedených poznatků. Aplikace je editorem technologických scén. Z jednotlivých prvků, modelů technologických celků, uložených v databázi, je možné složit technologickou scénu, nadefinovat v ní vhodné pohledy a uložit. Každému z vložených modelů je

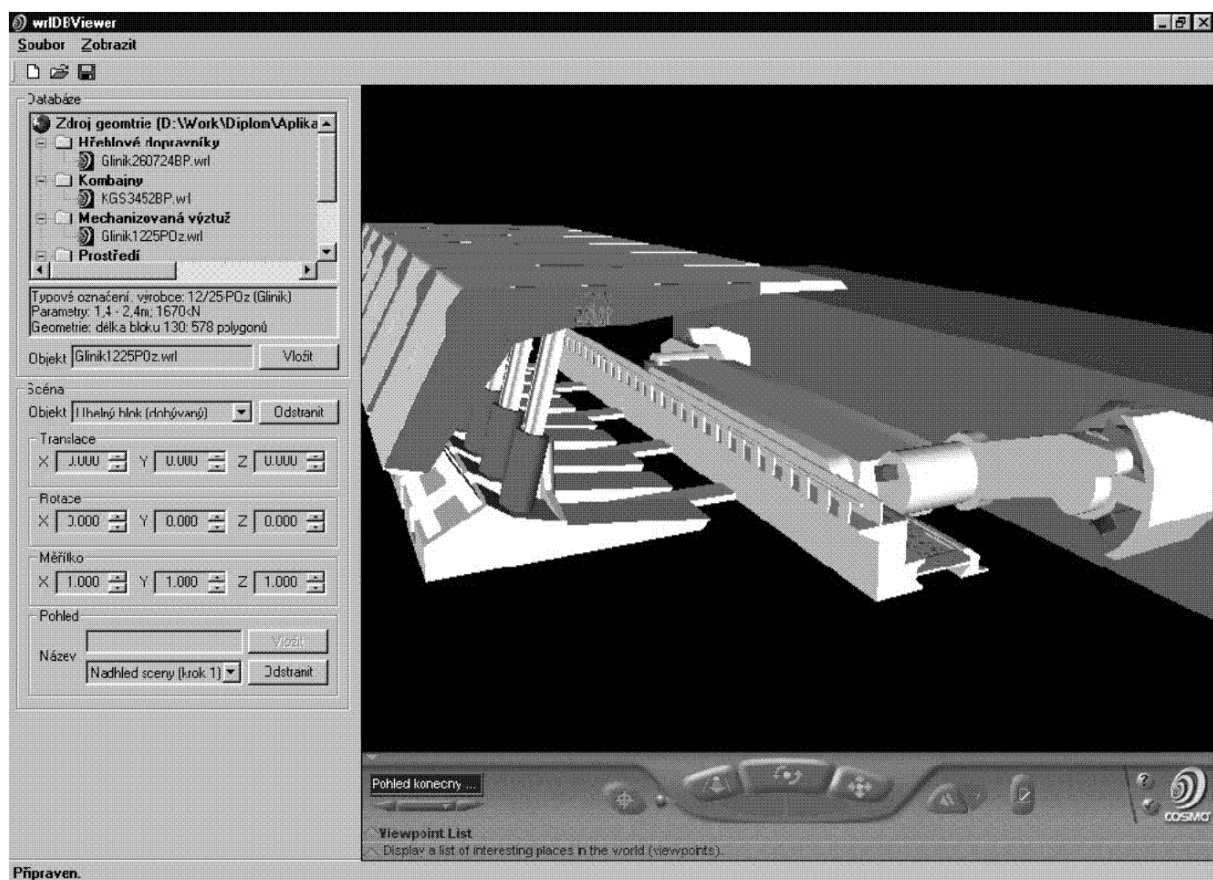
možné přiřadit jméno a tento model lze příslušnými způsoby transformovat (posouvat, otáčet a měnit měřítko) nebo jej ze scény odstranit. Databáze obsahuje i prvky prostředí (uhelný pilíř a další).

### Vlastní řešení zobrazení technologického procesu

Realizovaná aplikace je vytvořena pomocí Visual C++ a MFC. Pro začlenění VRML prohlížeče byla použita COM technologie. K propojení s databází bylo využito rozhraní ODBC, API sjednocující metodologii propojení uživatelské aplikace s databází bez závislosti na jejím výrobci i fyzickém umístění.

Během vytváření prázdné scény je vytvořen seskupovací uzel. Každý nový objekt je při vkládání do scény nejprve vygenerován ze zdrojového textu v databázi pomocí EAI funkce (Browser::createVrmlFromString), následně je zapouzdřena pomocí vstupní události (addChild) do transformačního uzlu (Transform), jehož reference je uložena v seznamu používaných uzlů pro případnou pozdější modifikaci. Takto transformovaný objekt je konečně začleněn do scény jako prvek seskupovacího uzlu. Podobným postupem jsou do scény zařazeny jednotlivé pohledy (CameraPosition). Jejich poloha – souřadnice posunutí a orientace jsou zjišťovány nestandardně pomocí rezervovaného ( a nedokumentovaného) rozhraní IUCPReserved.

Aplikační okno je rozděleno na zobrazovací prvek (komponenta VRML prohlížeče) a ovládací panel, který je dále rozdělen na dvě části: Databáze a Scéna.



Obr.10. Prostředí navržené aplikace.

Fig.10. Surrounding of the proposed application.

### Databáze

Databáze je uživateli prezentována ve tvaru hierarchické stromové struktury, kde každá větev představuje tabulku (v tomto případě druh technologického celku) a jednotlivé listy pak její položky (vlastní geometrii modelu). Označení položky má za následek zobrazení příslušného popisu a aktivaci tlačítka Vložit. K dispozici je také editační pole umožňující změnu původního názvu geometrie vkládané do scény. Toto je výhodné pro odlišení více vkládaných objektů stejného typu – několik hřeblových dopravníků, sekce mechanizované výztuže apod.



## Scéna

Scéna (její obsah) je zprostředkována listovacím polem, které obsahuje jména všech modelů vložených do scény. Výběrem určité položky lze realizovat funkce Odstranit a editovat Translace, Rotace a Měřítko. Tím lze provést dodatečnou úpravu objektů scény.

K definici stanovišť slouží ovládací prvky v části Pohled. Zadáním jména se aktivuje tlačítko Vložit. Výběrem již definovaného pohledu lze realizovat funkci Odstranit.

Realizovaný systém vychází z funkcionality prostředí VRML, realizuje propojení prohlížeče s databází vlastním uživatelsky definovaným logickým meziklánkem. Tento byl realizován proto, že dosud není standardně realizováno řešení v rámci VRML specifikace. Jisté návrhy již sice existují, stále však nejsou součástí prohlížečů. Tato řešení se týkají především rozšíření funkčních možností skriptů (založeno na JDBC) obsažených přímo ve zdrojovém textu scény o možnosti databázových manipulací. Skriptování v rámci scény se však nezdá být dostatečně silným nástrojem. Značným handicapem je zde omezení vstupních a výstupních operací směrem k uživateli. Za plnohodnotné řešení považují použití EAI. Toto se však může změnit s příchodem nové specifikace X3D. Zatím lze EAI použít téměř k jakémukoliv účelu. Důležité je zvládnutí programovacích technik i technologií využívajících COM-EAI.

## Závěr

Na základě analýzy technologického procesu hlubinného dolu byl sjednocen způsob popisu jednotlivých technologických subsystémů. Toto umožnilo využít podobné datové struktury pro jejich popis. Data uložená v databázi v jednotlivých tabulkách lze využít pro základní vizualizační úlohy pro zobrazení plně 3D technologických scén. Výhodou zvolené technologie je možnost takovou scénou procházet a zobrazovat si příslušné detaily optimálním způsobem.

Neznamená řešení ukazuje silný potenciál zvolené technologie, nicméně snaha o maximální rozsah i komplexnost scény vedou k určitým omezením. Vývoj VRML nepostupuje tak rychle, jak by si jeho autoři přáli a zdá se, že bude nezbytné ověřit i jiné (komerční) produkty. Jednou z možností je Multigen Creator spolu s Vega Prime.

Současné běžné počítače využívají periferní zařízení, která jsou konstruována pro práci s pouze dvourozměrnými objekty. Prezentace 3D scény na takových zařízeních je samozřejmě možná, nicméně v této oblasti existují speciální zařízení pro práci ve 3D. Jde o různé konstrukce zařízení pro prostorové vidění (např. náhlavní displej), datové rukavice pro interakci ve scéně i různé systémy trajler, které snímají reálnou pozici pozorovatele ve scéně. Těmito prostředky lze dále umocnit vnímání 3D scény a překročit hranici směrem k virtuální realitě.

## Použité zkratky

ANSI	American National Standards institute	ODBC	Open database Connectivity
API	Application Programming Interface	OLE	Object Linking and Embedding
ASP	Active Server Pages	SQL	Structured Query Language
CLOB	Character Long Object	VM	Virtual Machine
COM	Component Object Model	VRML	Virtual Reality Modeling Language
EAI	External Authoring Interface	W3C	World Wide Web Consortium
GUI	Graphical User Interface	X3D	Extensible 3D
HTML	Hypertext Markup Language	XML	Extensive Markup Language
JDBC	Java Database Connectivity		

## Literatura

- KODYM, O.: System for Storing and Presentation of Information about the Underground Mine. In Mine Planning and Equipment Selection '97. Ostrava, Czech Rep. VSB–Technical University Ostrava 1997. ISBN 90-5410-915-7.
- KODYM, O.: Representation of the technological process of the underground mine. In Proceedings The 14th International Conference on Automation in Mining. Helsinki : Helsinki University of Technology, 2001. vol. 14, s. 75-85.
- KODYM, O.: Usage of Virtual Reality in Control and Monitoring of the Underground Technological Process. In Mine Planning and Equipment Selection '02. Bouzov, Czech Rep. VSB–Technical University Ostrava 2002. ISBN 80-248-0127-2